Tackling uncertainty in security assessment of critical infrastructures
Dempster-Shafer Theory vs. Credal Sets Theory

Misuri, Alessio; Khakzad, Nima; Reniers, Genserik; Cozzani, Valerio

DOI
10.1016/j.ssci.2018.04.007

Publication date
2018

Document Version
Final published version

Published in
Safety Science

Safety Science 107 (2018) 62–76

Contents lists available at ScienceDirect

Safety Science

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

Tackling uncertainty in security assessment of critical infrastructures: Dempster-Shafer Theory vs. Credal Sets Theory

Alessio Misuri[a], Nima Khakzad[b,*], Genserik Reniers[b], Valerio Cozzani[a]

[a] Department of Civil, Chemical, Environmental, and Materials Engineering, University of Bologna, Bologna, Italy
[b] Safety and Security Science Group, Faculty of Technology, Policy, and Management, TUDelft, Delft University of Technology, The Netherlands

ARTICLE INFO

Keywords:
Security vulnerability assessment
Attack tree
Uncertainty modeling
Dempster-Shafer Theory
Bayesian network
Evidential network
Credal network

ABSTRACT

Securing critical infrastructures is a complex task. Required information is usually scarce or inexistent, and experts' judgments may be inaccurate and biased. In this paper, two methodologies dealing with data scarcity, imprecision, and uncertainty are presented: Evidential network and Credal network. Evidential network is a graphical technique based on Dempster-Shafer Theory to explicitly model the propagation of epistemic uncertainty among variables while Credal network is an extension of Bayesian network to deal with sets of probabilities, known as Credal sets, based on experts' judgments. Both methodologies constitute robust frameworks to account for high degree of imprecision on data, producing informative results despite the low-informative input. In the present study, the power in expressing uncertainty of these two methodologies have been showed, and their differences have been described through their application to a case study of security vulnerability assessment. Results demonstrate the substantial equivalence of the two methodologies in prognostic analysis, thus, an approximate updating procedure of Evidential network through equivalent Credal network has been proposed, to overcome the lack of possibility to compute updating in the context of Dempster-Shafer Theory.

1. Introduction

Since the 9/11 terrorist attacks, the concern about malevolent actions against critical infrastructures has remarkably grown. Indeed, before the tragedy of Twin Towers, the perception of risk was limited to unintentional events, as natural disaster or technical failures of critical systems (Baybutt and Ready, 2003). The discipline of security vulnerability assessment (SVA) is relatively young, and has been developed to provide guidelines and methodologies to highlight weaknesses potentially exploitable by adversarial agents to carry out high-consequences detrimental actions against critical assets. The chemical and process industry is one among the first industrial sectors where a number of methodologies were developed to mitigate the risk of security-related events (API (American Petroleum Institute), 2003; CCPS, 2003; Bajpai and Gupta, 2005). According to API RP-70 (API (American Petroleum Institute), 2003), the security risk is defined as the product of the magnitude of consequences caused by an attack, the probability that the attack will be carried out, and the success probability of the attack.

An attack's conditional probability of success has to be estimated considering the security system and its weaknesses. This step is commonly referred to as vulnerability assessment, and is a key step of the analysis. Vulnerability in security risk assessment refers to any weakness which can be exploited by a malevolent agent to gain access to an asset (API (American Petroleum Institute), 2003). Thus, elements to take into account in vulnerability assessment may include the location of the plant, the potential failure of physical protection systems, the equipment and its properties, and personnel practices. According to the majority of these methodologies, the determination of scenarios, and thus vulnerability assessments are based on semi-quantitative calculations, usually relying on experts' judgments. Multilateral competences and high expertise are needed because of the complexity of security issues. The required information embraces various fields, varying from technical to socio-political, whereas available historic data is scarce or even inexistent. Thus, the mission of security research is to develop methodologies able to provide reliable results despite the high level of uncertainty and subjectivity characterizing this field.

Attempts to efficiently deal with the inherent uncertainty of parameters in SVA have been made, usually based on probabilistic techniques and experts' judgement. Argenti et al. (Argenti et al., 2016) propose to adopt Bayesian network (BN) to model the effectiveness of security systems in process installations. Fakhravar et al. (Fakhravar et al., 2017) propose a vulnerability analysis based on attack trees (ATs)

* Corresponding author at: Jaffalaan 5, Delft 2628BX, Netherlands.
E-mail address: n.khakzadrostami@tudelft.nl (N. Khakzad).

https://doi.org/10.1016/j.ssci.2018.04.007
Received 16 November 2017; Received in revised form 3 April 2018; Accepted 13 April 2018
Available online 24 April 2018

Nomenclature

SVA	security vulnerability assessment
IED	improvised explosive device
PPS	physical protection system
AT	attack tree
FT	Fault tree
BN	Bayesian network
DAG	directed acyclic graph
CPT	conditional probability table
CS	Credal Set
CN	Credal network
DST	Dempster-Shafer Theory
ET	Evidence Theory
EN	Evidential network
CBT	Conditional belief table
BBA	basic belief assignment
Pls	Plausibility function
Bel	Belief function

and subsequent mapping through an innovative time-based BN. Clearly, BN is an attractive tool for this aim, because it is able to gather variables of various nature, and to probabilistically depict dependencies, intuitively expressing uncertainty.

A criticism against the use of probabilities is that they may not be easy to be assessed as point values. On the other hand, it may be more natural for experts to represent their opinion through comparative judgments, intervals of probability, or degrees of belief. Moreover, even if experts were able to directly convert their statements in probability values, numbers would be affected by epistemic uncertainty due to the impossibility of practically obtaining some information or clearly shaping dependencies. The objective of this paper is to specifically examine the applicability of two methodologies to deal with epistemic uncertainty and imprecision in SVA, and then comparing their features and outlining the differences between the two approaches. The present paper proposes a comparison between Dempster-Shafer Theory (DST) and Credal network (CN). DST allows to explicitly model the propagation of epistemic uncertainty among the variables of a system, depicted as an Evidential network (EN). This methodology may offer an intuitive framework to use low informative judgments to obtain reliable outcomes, keeping track of effects of the vagueness of input information on the obtainable results, through simple mathematical functions. On the contrary, CN is conceptually more similar to BN, and is based on the specification of sets of probabilities rather than on point values. These sets can be defined through geometrical figures whose edges can be directly reconstructed starting from comparative judgments, through mathematical procedures based on standardized interpretation of natural language. Both methodologies are suitable to produce robust outcomes from low informative input data. Therefore, these methodologies may be of great application to the field of SVA since SVA usually suffers from qualitative judgments, biased subjective data, and imprecise information.

As such, EN may allow to evaluate the quality of results, producing optimistic and pessimistic estimations of vulnerability via plausibility and belief concepts while CN may allow the specification of probabilities of primary events in a more robustly in the form of set of possible values. That being said, the novelty of the present study lies in the application of EN and CN to uncertainty modelling which has been unprecedented not only in SVA but also safety risk assessment.

After revisiting the background of AT, BN, CN, and DST in Section 2, the methodologies are applied to a case study in Section 3. Section 4 is reserved for the discussion of results, comparing the methodologies, and pointing out their shortcomings. Conclusions are reported in Section 5.

2. Background

2.1. Attack tree

AT is a hierarchical graphical framework to model attacks against a system, given some security constraints (Brooke and Paige, 2003). ATs have been outlined for the first time by Schneier (Schneier, 1999, 2000) in the field of informational technology. In his definition, ATs are powerful tree-shaped representations and offer a clear view of the security system and its components. For a solid mathematical definition of AT, the reader is referred to (Gribaudo et al., 2015). For the scope of this paper, it is sufficient to indicate that ATs are analogous to Fault trees (FTs) (US Nuclear Regulatory Commission, 1981) but that they are used for security risk assessment rather than of safety risk assessment (in case of FTs). AT makes it possible to model failure sequence of countermeasures, or success of intermediate steps of attack via AND and OR gates, shedding light on the vulnerabilities of security systems. It is worth noting that there are some differences between ATs and FTs. For example, the FT's structure is linked to the architecture of the system, so it is not mutable unless there is a change in the system, while AT's structure depends on the effect of countermeasures on attackers' preferences, and so it may change radically after each improvement of the security system (Gribaudo et al., 2015).

Fig. 1 depicts an AT for opening a safe. As shown in the figure, leaf nodes are various types of attacks, and the root node "Open Safe" is the goal the attacker wants to achieve. Actually, leaf nodes can also represent states of elements of the security system, and so their effectiveness to stop an attack. As can be noted from Fig. 1, originally the semantic rules adopted by Schneier (Schneier, 2000) are opposite to the ones usually adopted in FT analysis (i.e., the "Open Safe" node should be a leaf node depicting a top event). Since in the field of process engineering FTs are widespread and well-known tools, for the rest of the paper the semantics of FT was adopted also for ATs. Thus, in the rest of the paper the basic events (i.e., types of attack, or state of elements of security system) will be labelled as root nodes, while the vulnerability (i.e., the probability of success of an attack) will be the top event, or leaf node. This notation is more intuitive considering that ATs in this paper will be mapped into directed acyclic graphs similar to BN.

ATs can be used both for qualitative and quantitative analysis of the security system. Qualitative analysis is performed based on the graph depicting dependencies, and allows experts to brainstorm knowledge,

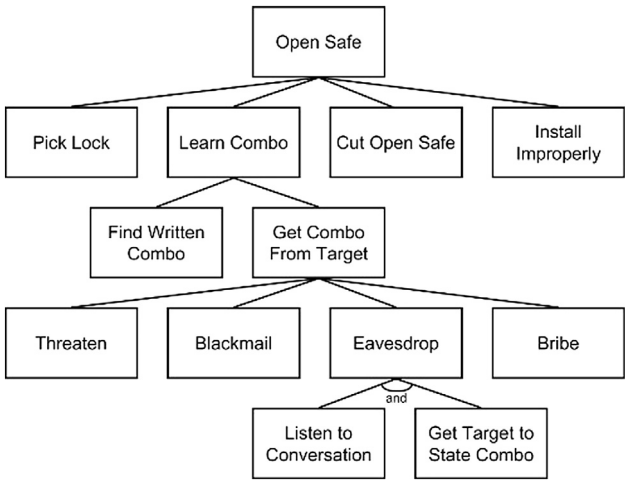

Fig. 1. AT for opening a safe (Schneier, 2000).

and to enhance discussion (Ingoldsby, 2013). In order to conduct a quantitative analysis, experts have to assess probabilities to each node, and then find an efficient way to evaluate the model, which may be time-demanding for systems with high number of components. As for FTs evaluation (Bobbio et al., 2001), experts can map the ATs in BN and compute the probability of successful attacks (Fakhravar et al., 2017; Gribaudo et al., 2015).

2.2. Bayesian network

BN is a widespread probabilistic tool for eliciting knowledge and for dealing with uncertainty (Pearl, 1988). BN synthetically represents relations among random variables through a directed acyclic graph (DAG) composed of nodes and arcs. The weight of dependencies is locally defined with conditional probability tables (CPTs). The joint probability distribution of a set of random variables $X = \{X_1,X_2,...,X_n\}$ can be easily computed, thanks to *d*-separation criteria (a set of criteria to determine whether connected nodes are independent of each other given the state of another node; for further details refer to (Pearl, 1988; Charniak, 1991), and the chain rule considering conditional probabilities of each variable given its immediate parents:

$$P(X) = P(X_1,X_2,...,X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i)) \tag{1}$$

where $pa(X_i)$ is called parental set of X_i. For example, considering the BN in Fig. 2, the joint probability distribution is $P(X_1,X_2,X_3,X_4) = P(X_1)\cdot P(X_2|X_1)\cdot P(X_3|X_1,X_2)\cdot P(X_4|X_2,X_3)$.

One of the major features of BN is that it is possible to update probabilities given new evidence through Bayes' Theorem (Eq. (2)):

$$P(X|E) = \frac{P(X)\cdot P(E|X)}{\sum_{X/E} P(X)\cdot P(E|X)} \tag{2}$$

where $P(X|E)$ is the updated (posterior) joint probability given evidence E, and $\sum_{X/E} P(X)\cdot P(E|X)$ is the summation over X except of E (Charniak, 1991).

BN has been widely employed in system safety and reliability engineering. Bobbio et al. (2001) and Khakzad et al. (2011) proposed to map FT into BN for dependable systems; likewise, Khakzad et al. (2013) developed an algorithm to map bow-tie diagram to BN to model dependencies and probability updating. However, few works based on BN are available in the field of security assessment (Argenti et al., 2016; Fakhravar et al., 2017; Van Staalduinen et al., 2017).

In this study the GeNIe software (GeNle Modeler) is used to draw and compute the developed BNs.

2.3. Credal Sets Theory and Credal network

Credal Sets Theory is a complete probabilistic theory based on Credal sets (CSs), that is, closed convex sets of probabilities to express knowledge imprecision (Corani et al., 2012). This theory is equivalent to Walley's theory of imprecise probabilities (Walley, 1991). CSs adopted in this theory can be depicted as polytopes, where each inner point has a valid probability mass, and can be obtained computing the convex hull of a finite number of probabilities, called vertices (Cozman, 2000). A CS for a random variable X_i is labelled $K(X_i)$, while the set comprising its extreme points is denoted by $ext[K(X_i)]$. In Fig. 3, an example from (Piatti et al., 2010) has been illustrated: a CS over a ternary categorical variable S, whose set of possible values is $\Omega_S = \{W,D,L\}$, is represented in barycentric coordinates as defined in (Walley, 1991). This system of coordinates is based on the property that the sum of probabilities of the three possible outcomes, for each point is necessarily 1.0. That is why probability values are represented as points inside the triangle of unitary height. The CS in Fig. 3 is coloured in red, and has four vertices:

$$ext[K(S)] = \left\{[0.5,0.3,0.2],[0.5,0.25,0.25],[0.4,0.4,0.2],\left[\frac{1}{3},\frac{1}{3},\frac{1}{3}\right]\right\} \tag{3}$$

CSs can be defined univocally by explicit enumeration of vertices. The vertices can be reconstructed starting from linear sets of inequalities, through *lrs*, a well-known algorithm for Reverse Search Vertex Enumeration (Avis and Fukuda, 2000, 1992). For example, the CS presented in Fig. 3 can be reconstructed through *lrs,* starting from the following set (Piatti et al., 2010):

$$\begin{cases} P(W) \leqslant 0.5 \\ P(W) \geqslant P(D) \\ P(D) \geqslant P(L) \\ P(L) \geqslant 0.2 \\ P(W) \geqslant 0 \\ P(D) \geqslant 0 \\ P(L) \geqslant 0 \\ P(W) + P(D) + P(L) = 1 \end{cases}$$

The last line of the set is an equation rather than an inequality, and represents the condition that the sum of the probabilities among all the possible outcomes has to be one (i.e., the only accepted outcomes of the variable S are *W, D,* and L). The possibility to represent the knowledge as a set of inequalities can be useful because experts can easily express their uncertainty about states of variables through comparative judgments, and starting from their arguments it is possible to reconstruct a limited set of probability masses. Furthermore, Walley (1991) proposes a verbal scale to directly translate judgments into linear constraints. For example, if an outcome x_i of a variable X is said to be "*very unlikely*", it means that $P(x_i) \leqslant 0.25$, while if x_i is said to be "*quite probable*", will be $P(x_i) \geqslant 0.6$.

As said before, CSs are geometrically depictable as polytopes, however, in practical applications we are interested in bounds of probability provided by them. *Lower* and *upper probabilities* obtainable for a variable X_i are respectively defined by Eqs. (4) and (5):

$$\underline{P}(X_i) = \min_{P(X_i)\in ext[K(X_i)]} P(X_i) \tag{4}$$

$$\overline{P}(X_i) = \max_{P(X_i)\in ext[K(X_i)]} P(X_i) \tag{5}$$

For example, the CS presented in Fig. 3 leads to $\underline{P}(D) = 0.25$, $\overline{P}(D) = 0.4$, $\underline{P}(W) = \frac{1}{3}$, $\overline{P}(W) = 0.5$, $\underline{P}(L) = 0.2$, $\overline{P}(L) = \frac{1}{3}$.

It is worth noting that CSs over binary variables are intervals of probability whose extremes are exactly $[\underline{P}(X_i),\overline{P}(X_i)]$: actually, probability intervals and p-boxes are special cases of CSs (Walley, 1996),

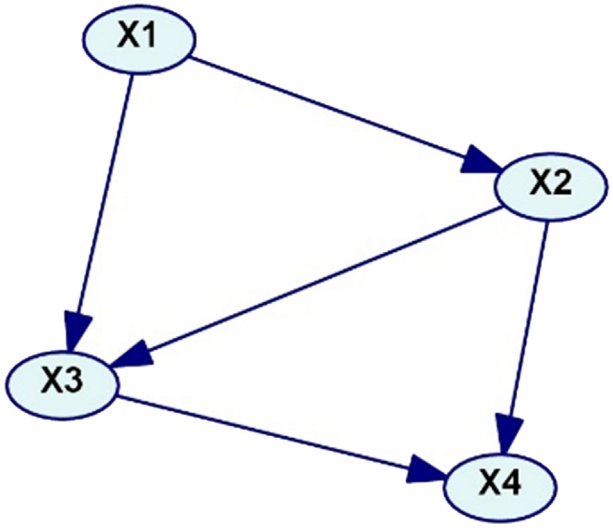

Fig. 2. A simple BN with four nodes.

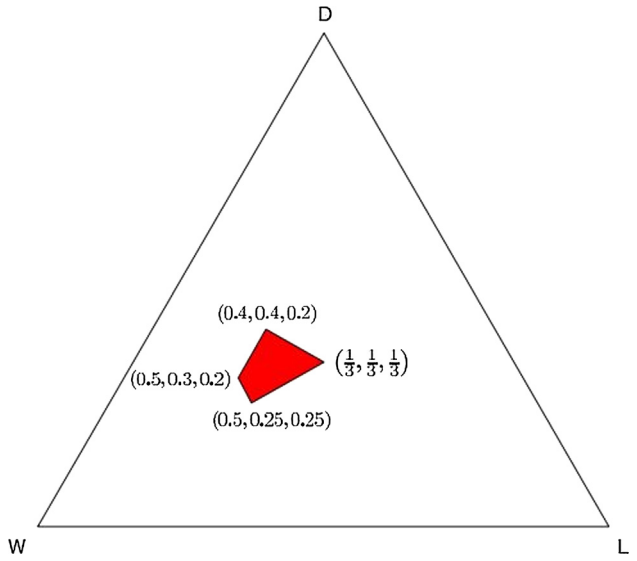

Fig. 3. Barycentric representation of a Credal set over a ternary variable (Piatti et al., 2010).

and represent aleatory uncertainty among the probability values. Thus, each value inside CSs, as well as inside intervals, is a valid probability mass.

After the definition of CSs, it is possible to define a Credal network (CN), that is, a BN whose parameters can vary according to closed convex sets. A CN over a set of random variables $X = \{X_1,X_2,...,X_n\}$ is a DAG where dependencies among variables are defined by a set of conditional CSs as $K(X_i|pa(X_i))$ for i = 1 … n. In analogy with BN, it is possible to define a joint CS as follows:

$$K(\boldsymbol{X}) = CH\{P(X)\colon P(x) = \prod_i P(x_i|pa(X_i))\}$$
$$where\ P(X_i|pa(X_i) \in ext[K(X_i|pa(X_i))],\ i = 1...n\} \tag{6}$$

where *CH* is the convex hull operator, applied to the probabilities computed for the combination of all the vertices of all the conditional CSs. The joint $K(\boldsymbol{X})$ defined in Eq. (6) is called strong extension of the CN. For further details about the meaning of strong extension, and other possible extensions the reader is referred to (Cozman, 2000; Antonucci and Zaffalon, 2008; Couso et al., 1999). In the following, the definition provided by Eq. (4) was applied, since literature focuses more on strong extension than on other definitions of joint CS. Furthermore, strong extension is the only which is possible to compute through the available software. Roughly speaking, the strong extension represents a set of BN obtained by visiting each possible combination of vertices of all the conditional CSs (Walley, 1996).

CN takes the advantage of a generalization of Bayes' Theorem to compute posterior probability bounds, with respect to the strong extension $K(\boldsymbol{X})$. Given the set of vertices of $\{P_j(\boldsymbol{X})\}_{j=1}^{v} \equiv ext[K(\boldsymbol{X})]$, the posterior lower probability $\underline{P}(X_q = x_q|X_e = x_e)$, given the evidence $X_e = x_e$, can be computed according to Eq. (5):

$$\underline{P}(x_q|x_e) = \min_{j=1,...,v} \frac{\sum_{x_M} \prod_i P_j(x_i|pa(x_i))}{\sum_{x_M,x_q} \prod_i P_j(x_i|pa(x_i))} \tag{7}$$

where $x_m \in X_M \cdot X \setminus (\{X_q\} \cup X_E)$, with X_M is the set of variables excluding X_q and X_E, and the values of x_i and $pa(x_i)$ are consistent with (x_q,x_M,x_e) (Corani et al., 2012; Antonucci, 2008). To obtain the posterior upper probability it is sufficient to replace the minimum operator with the maximum. Despite the task of computing posterior bounds is generally NP^{PP}- complete (Corani et al., 2012), that is, it belongs to a class of problems whose solution cannot be efficiently located, two available tools implementing different algorithms were developed. The first tool is JavaBayes, a software tool originally designed for robustness analysis in BN (Cozman), implementing an approximate algorithm (Cano et al., 1994), while the second is GL2U, a software tool specifically designed for CN (Sun, 2008), implementing a more accurate inference algorithm (Ide, 2004). Some applications of CN to deal with imprecise knowledge are reported in the literature (Antonucci et al., 2004, 2013).

2.4. Dempster-Shafer theory and Evidential network

Dempster-Shafer Theory (DST), also known as Evidence Theory (ET), has its starting points in the studies of Dempster (1967) and of Shafer (1976). Basically, it can be interpreted as a generalization of Bayesian probability, assigning a number between 0 and 1 to degree of belief supporting a certain proposal. It is worth to specify that even if this assessment may resemble probability assignment, it has not a probabilistic meaning. This above-mentioned procedure is called basic belief assignment (BBA) and is made defining a belief mass function (or Shafer's basic probability structure) M: $2^{\Omega} \rightarrow [0,1]$, which verifies:

$$M(\varnothing) = 0 \tag{8}$$

$$\sum_{s_i^X \in \Omega} M(X = s_i^X) = 1 \tag{9}$$

where Ω is the set of disjoint states of M, called frame of discernment of focal elements of M. The fact that M is defined starting from the powerset of the frame of discernment, allows softening the probability framework in a more flexible structure. It is possible to allocate a quantity supporting the focal element $\{s_i^X,s_j^X\}$, which means that the variable X can both be in the state s_i^X or s_j^X and we are not able to determine the amount of masses to attribute to each state. This is a method to characterize epistemic uncertainty about the state of the variable, and the focal element $\{s_i^X,s_j^X\}$ is called epistemic state. For example, considering a typical binary variable, a typical frame of discernment may be Ω = {*True*,*False*}, the powerset where we define the BBA is $2^{\Omega} = \{M(X = \varnothing),M(X = True),M(X = False),M(X = \{True,False\})\}$, where the last element is the epistemic state.

Once the BBA has been realized, it is possible to obtain information about the states of the variables, computing two functions called *Plausibility* and *Belief functions*. The *Plausibility function (Pls)* expresses the plausibility of a state, given the evidence of the masses which contribute to its realization. Formally *Pls*: $2^{\Omega} \rightarrow [0,1]$ is defined by Eq. (10):

$$Pls(X = s_i^X) = \sum_{s_j^X | s_i^X \cap s_j^X \neq \varnothing} M(X = s_j^X) \tag{10}$$

The *Belief function (Bel)* expresses how much the proposal $X = s_i^X$ is believable, given the evidence of the masses of the frame of discernment. Formally *Bel*: $2^{\Omega} \rightarrow [0,1]$ is defined by Eq. (11):

$$Bel(X = s_i^X) = \sum_{s_j^X | s_j^X \subseteq s_i^X} M(X = s_j^X) \tag{11}$$

Recalling the previous example, if we assign $M(X = True) = 0.1,M(X = False) = 0.8,M(X = \{True,False\}) = 0.1$, it is easy to compute $Bel(X = True) = 0.1$, $Bel(X = False) = 0.8$, $Pls(X = True) = 0.2$ and $Pls(X = False) = 0.9$. Thus, *Pls* and *Bel* functions are needed to redistribute epistemic uncertainty among possible states of variables, without considering any unnecessary additional assumption. According to Shafer:

$$Bel(X = s_i^X) \leqslant P(X = s_i^X) \leqslant Pls(X = s_i^X) \tag{12}$$

Given Eq. (12), it is possible to associate an interval-valued probability to an assumption $X = s_i^X$, with minimum and maximum probabilities as *Bel* and *Pls,* respectively (Cheng, 2000). Fig. 4 helps clarify the quantification of the uncertainty about a certain hypothesis given a BBA. It is also possible to reconstruct BBAs from *Pls* and *Bel* functions,

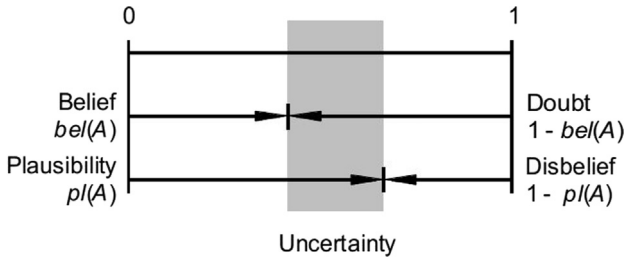

Fig. 4. Quantification of uncertainty through Bel and Pls functions. From Rakowsky (2007).

through a Möbius Transformation (Smets, 2002).

This theory can be applied to describe complex systems, and the involved variables. This scope leads to the foundation of Evidential network (EN), which is a DAG where dependencies among its nodes are defined by a set of belief distributions (Simon et al., 2008). Despite its similarity with BN, in EN, relations between variables are not probabilities, but belief masses. Practically, CPTs are replaced with CBTs (*Conditional Belief Tables*). Furthermore, additional nodes to represent *Bel* and *Pls* of variables of interest are added to the network.

Although DST is a flexible generalization of Bayesian theory able to represent information obtained directly from experts' judgments (i.e., degrees of support to hypothesis), only a few toolboxes exist such as IPP Toolbox (IPP Toolbox), and TBMLAB (Smets, 2004). This is the main reason why Simon et al. (2008) propose to employ BN algorithm to compute belief propagation in EN. Furthermore, the same authors propose to adopt EN to map FT when there is imprecision on data and on experts' judgments (Simon and Weber, 2009), according to the methodology for modelling uncertainty previously defined by Guth (1991). Other applications in the fields of reliability and safety can be found in Kay (2007) and Limbourg et al. (2007), while Benavoli et al. (2009) propose an EN for threat assessment in military operational research.

3. Results

3.1. The case study

We applied the two methodologies of interest to map ATs developed for the illustrative case study. The chemical storage plant whose premises are highlighted in Fig. 5 was chosen as a target for SVA. In Fig. 5 different assets can be distinguished. The main storage farm area is composed of eight tanks (of different volumes), and a loading dock, where ships can charge and discharge chemicals and materials. For the sake of simplicity, only a single attack made with improvised explosive device (IED) was considered, which could be detonated with a remote controller. Two possible intrusion paths were considered, that is, via ground, and via water from the area of the loading dock. The loading dock itself can be considered as potential target. However, according to a rough estimate of relative attractiveness of targets, we decided to choose the storage farm area as the target, due to the high quantity of hazardous materials stored, and the high visibility of the tanks.

Indeed, the aim of this study is to compare two robust methodologies to assess uncertainty and deal with imprecision, highlighting their features, rather than to conduct a complete SVA. In Fig. 5 a schematic representation of the two possible intrusion paths available to adversaries to reach the tank area have been denoted by arrows.

An assumption considered is that if perpetrators choose to carry out an attack via water, they have to complete the intrusion by a short secondary path via ground to reach the storage farm area, to set the explosive, and then regress before remotely detonating it.

This assumption implies that the attack will be successful only if perpetrators manage to regress to a safe distance from the storage plant. This assumption should be softened considering different kinds of perpetrators such as suicidal bombers who do not mind saving their lives to consider an attack successful (Bhashyam and Montibeller, 2016).

Considering the storage farm, two simple ATs have been developed, one for each path to get into the plant premises. In this study, for the sake of simplicity time has not been considered as a variable, even if there are similar applications suggesting that the intrusion process should be considered a timed sequence of actions. For example, Fakhravar et al. (2017) developed dynamic attack trees with priority AND (PAND) and Sequential Failure (SEQ) Gates, where the sequence of actions matters. Although considering AND-Gates instead of more complex PAND-Gates is a simplification, the results obtained will be conservative because PAND is a subset of AND (Gribaudo et al., 2015).

Two ATs, respectively depicting the intrusion processes via ground and via water, are displayed in Figs. 6 and 7.

As can be noted from Figs. 6 and 7, the right part of the two ATs is the same. Conceptually, perpetrators need to position an improvised explosive device ("IED") at the storage farm and regress ("Regress") before detonating the IED. The main difference between these graphs is related to the branches on the left, that is, the intrusion process. In Fig. 6 the left branch describes the situation when perpetrators decide to attack via ground. They may choose to force the "Main Gate" or to open an access on the outer fence "First Fence". After that, they need to penetrate "Patrol" to overcome the "First Security Layer". At this point, "CCTV", and an additional fence "Second Fence" have been considered. If perpetrators manage to elude both, the intrusion process "Intrusion" is considered successful.

In Fig. 7, the intrusion branch reports the path to be followed if perpetrators decide to carry out an attack via water. "Patrol", and "Docking Barriers" have been considered as the first security layer. After penetrating both, perpetrators have to continue via ground, so they have to penetrate the same "CCTV", and "Second Fence" considered in the former AT.

This means that there are dependencies between the two ATs due to the presence of shared physical protection systems (PPSs). Such dependence cannot be outlined directly in ATs, but has to be considered during calculations.

According to recent works of Bobbio et al. (2001) in the field of reliability analysis, Gribaudo et al. (2015) in the field of computer security, and Khakzad et al. (2011, 2013) in the field of process safety, mapping trees in BN allows a more efficient computation of probabilities, accounting for dependencies, and other advantages like probability updating and adapting.

Fig. 8 depicts the BN equivalent to the merged ATs, where the nodes

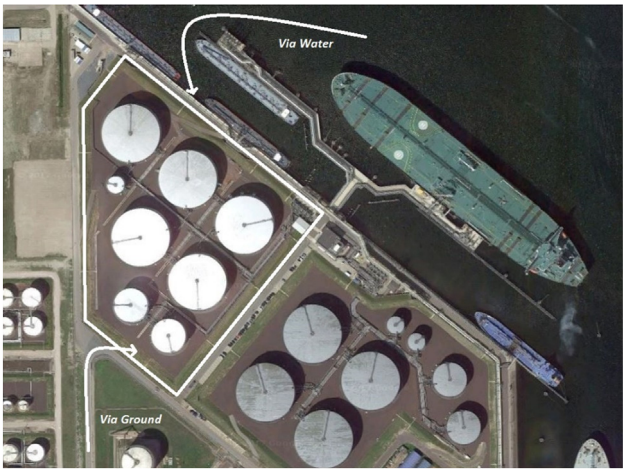

Fig. 5. Case study: premises of the storage farm are outlined in white. The two intrusion paths considered, "Via Ground" and "Via Water", are reported as white arrows.

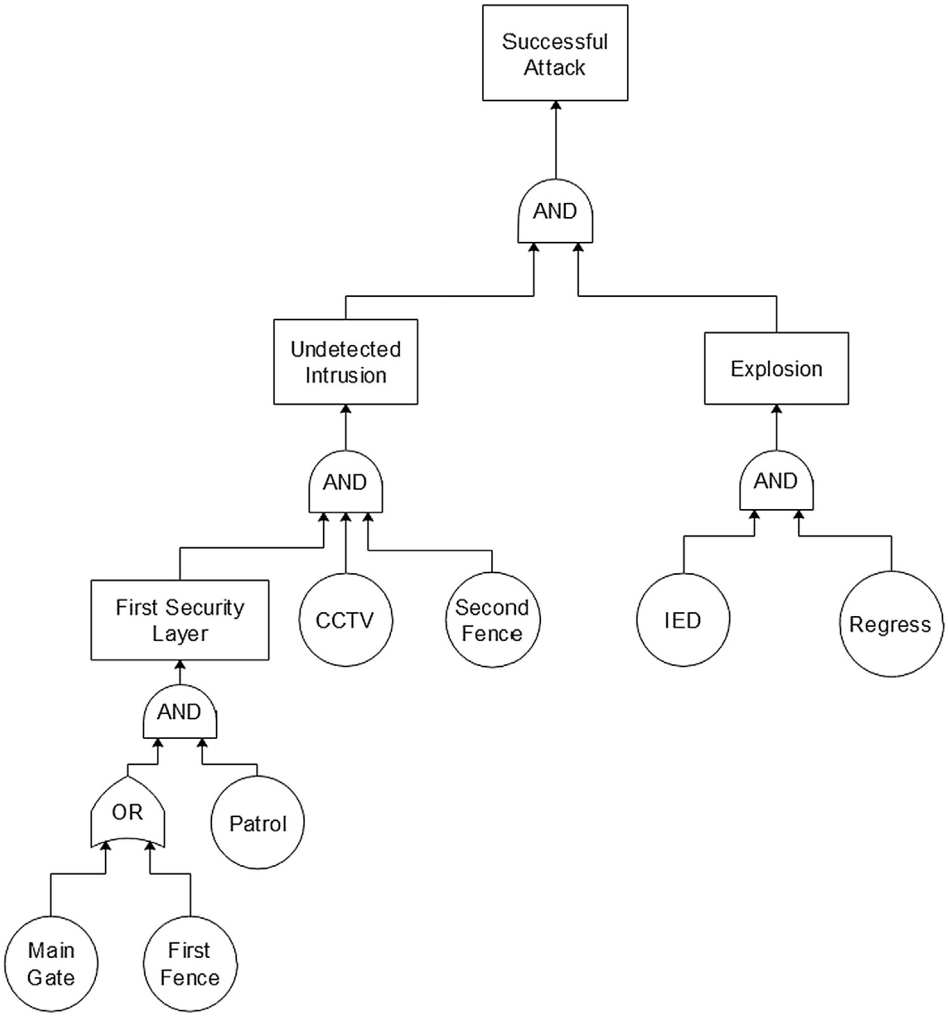

Fig. 6. Attack tree for intrusion and attack process via ground (In Fig. 6, we considered the possibility of attack through the "Main Gate" as it was the case for the attack to the Air Products Gas Company in Saint-Quentin Fallavier: https://www.ict.org.il/Article/1430/Saint-Quentin-Fallavier-Attack).

"Explosion", "CCTV", "Second Fence", and "Patrol" are shared by the two attack scenarios. In the merged mapping, a single leaf node "Attack" has been considered.

All variables in Fig. 8 are binary. Each variable can be in the state of "success" or "failure". Nodes corresponding to AND and OR operators have specific CPTs (Bobbio et al., 2001). Probabilities of the root nodes have to be assessed from data, or experts' judgments as point values, and then computed through algorithms available for BN.

In the following subsections the ATs are mapped through the two methodologies, in order to compare results, and usability of EN compared to CN.

3.2. Converting Attack trees to Evidential network

The main weakness related to traditional approach based on BN is that point values of probabilities are needed. The root nodes of the AT should be quantified probabilistically, but usually there is a significant lack of data and information. Thus, experts' judgments are the most used source of data (Nai et al., 2009). Given these premises, it seems clear that estimates of likelihood of events in SVA are inherently affected by epistemic uncertainty: data are affected by incompleteness, incoherency and sometimes are hard to be determined experimentally. EN may be employed alternatively as a framework to obtain robust yet less precise results from low informative data sources; it is because it is possible to associate part of the belief to the epistemic state (see Section 2.4). The epistemic state allows to define the amount of information which is not possible to associate with any of the other states. *Pls* and *Bel* functions can then be used to reconstruct confidence intervals for the states of the variables of interest.

The approach of Simon et al. (2008), Simon and Weber (2009) has been applied to convert the BN of Fig. 8 to the EN of Fig. 9. The methodology is based on the transformation of each binary variable of the BN in Fig. 8 into a ternary Dempster-Shafer structure in the EN in Fig. 9, in which the first two states correspond to the same states of the node in the BN while the third state is the epistemic state. Having the EN constructed this way, the BN standard inference algorithms can readily be used to compute belief propagation in the EN.

It is worth remarking that EN should not be interpreted as a probabilistic network since the values represent the degrees of support (belief) rather than probabilities. It is clear that the structure of the EN in Fig. 9 is very similar to the BN's in Fig. 8, yet augmented with "Pls" and "Bel" nodes. These two additional nodes are defined for explicit computation of *Pls* and *Bel* functions as the upper and lower bound probabilities, respectively, of the states of variables of interest (in our case, the Pls and Bel functions of the "Success" state of the primary events nodes).

The variables "Patrol", "Docking_Barriers", "Main_Gate", "First_Fence", "Patrol", "CCTV", "First_Security_Layer", and "Second_Fence" represent the PPSs from attackers' point of view, and so the state of the variables in the EN describe the attacker's ability to

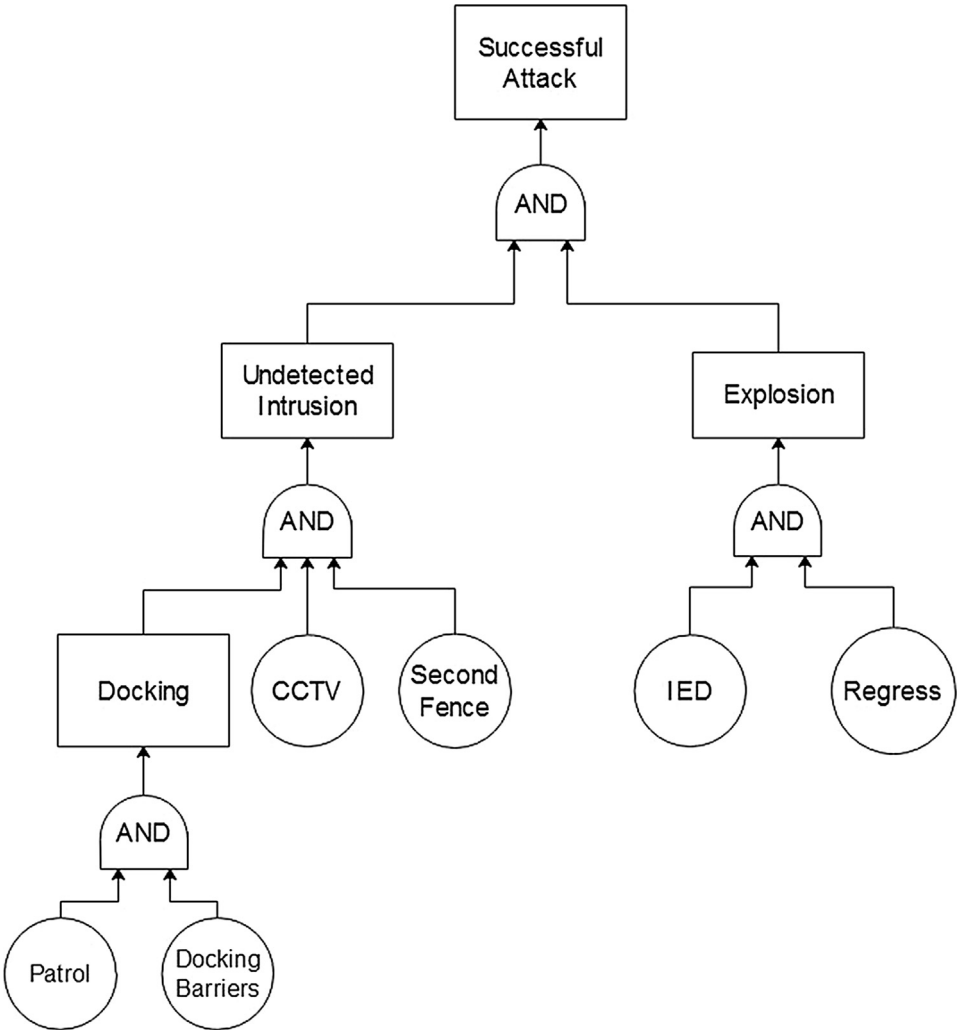

Fig. 7. Attack tree for intrusion and attack process via water.

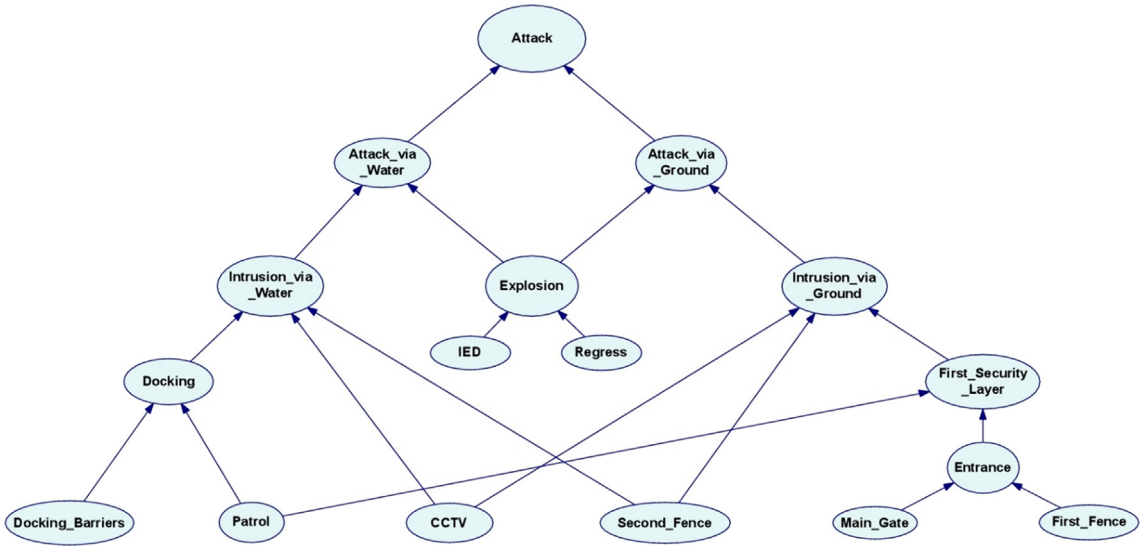

Fig. 8. ATs merged together in a BN. The central part comprises the shared variables.

overcome the security barriers, and proceed the attack. For example, "Patrol" = "Success" means that perpetrators manage to penetrate the patrol.

The variables "Docking", "Intrusion_via_Water", "Intrusion_via_Ground", "IED", "Regress", "Explosion", and "Entrance" represent steps the perpetrators have to successfully achieve to carry out an attack. For example, "Regress" = "Failure" means that perpetrators were not able to regress after setting the IED near the tanks.

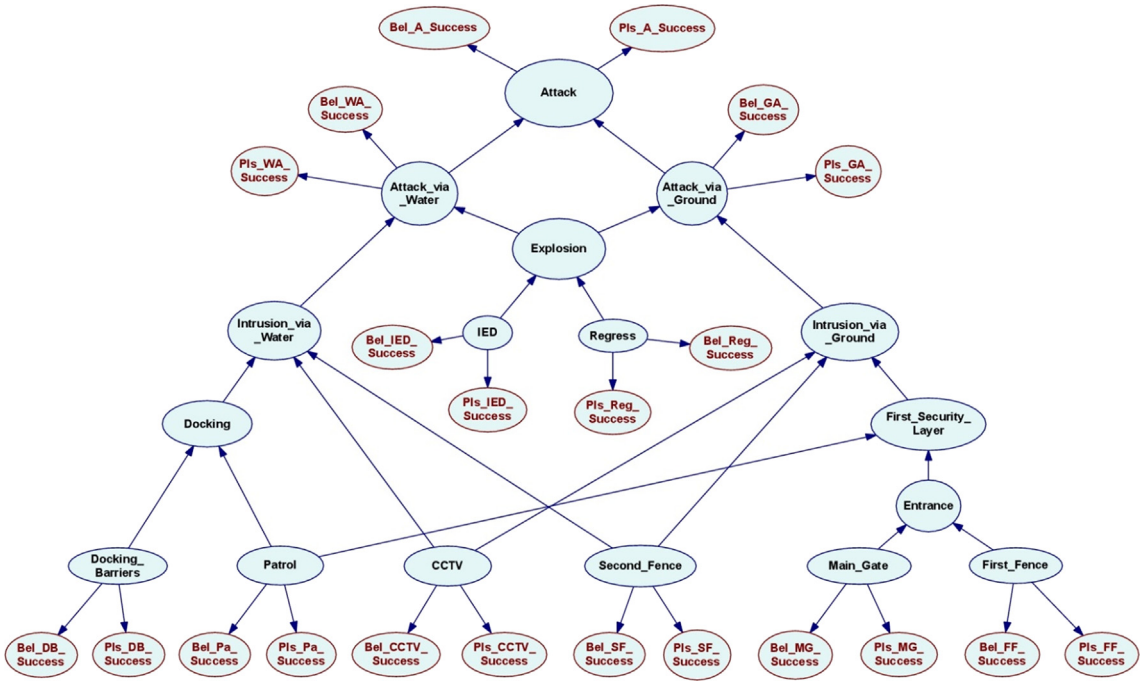

Fig. 9. EN of attack scenarios.

The variables "Attack_via_Water", "Attack_via_Ground", and "Attack" depict the success or the failure of the attack. That is, "Attack_via_Water" = "Success" means that the attack via water is successful, while "Attack" = "Success" means that an attack, either via ground via water, or both results in a success.

The methodology proposed by Simon et al. (2008), Simon and Weber (2009) has been applied to define CBTs of AND and OR gates. For example, "Docking" is an AND gate, so it can result in "Success", only if the states of both its parents "Patrol", and "Docking_Barriers" are "Success". This means that attackers have to penetrate the patrol and to overcome the docking barriers to successfully dock near the premises of the plant. According to that, the CBT of this variable with respect to the parents are reported in Table 1.

To define OR nodes, the same logic has been followed. For example, considering the "Attack" node, which is an OR node, its state can be "Success", if at least one of its parents ("Attack_via_Ground" and "Attack_via_Water") is "Success". Table 2 reports the CBT for this variable.

The nodes in red represent the *Bel* and *Pls* functions are computed for every output of the parent variables. For example, "Bel_A_Success", and "Pls_A_Success" are nodes reporting the values of *Bel(Attack = "Success")*, and *Pls(Attack = "Success")*. The CBTs proposed to compute their values are, respectively, reported in Tables 3 and 4.

The "Pls" and "Bel" nodes have been provided for attack nodes "Attack_via_Water", "Attack_via_Ground", and "Attack" as well as primary events. This choice was made because the presence of a high number of nodes may affect the readability of the graph. Furthermore, the values of *Bel* and *Pls* functions are of interest only for some variables. Indeed, according to Eq. (12) presented in Section 2, it is possible to directly reconstruct probability bounds from *Bel* and *Pls*, and this is useful to estimate the possibility of success of an attack, as an indication of the vulnerability of the security system. With respect to the basic events "Patrol", "Docking_Barriers", "CCTV", "Patrol", "Main_Gate", "First_Fence", "Second_Fence", "IED", and "Regress", red nodes have been provided because they report the probability intervals used as CSs in CN application presented in Section 3.3. They are not strictly necessary, but allow to directly show the probability bounds needed to compare the EN with the CN presented in the next subsection.

For illustrative purposes only, a priori belief mass assignments for each primary node were chosen as in Table 5.

Given these input data, forward analysis of belief propagation through GeNIe software was performed. Since the quantification of vulnerabilities is of interest, the results are the values of *Bel* and *Pls*, related to the state "Success" of the three attack nodes. Results obtained are reported in Table 6.

These values can be used directly as estimates of vulnerability. According to Table 6, a single value representing the vulnerability, i.e., the probability that, given an attack, it will be successful and will cause damage to the storage area, cannot be determined exactly. However, according to definition of *Belief* and *Plausibility functions*, we can define the former as the minimum degree of belief that the plant is vulnerable to an attack, while the latter is the maximum vulnerability.

For example, the minimum belief that an "Attack_via_Ground" will be successful is equal to Bel (X = "Success") = 1.24e − 3. If we consider also the uncertainty, that is, the belief mass associated to the epistemic state "Success, Failure", the vulnerability to an "Attack_via_Ground" would be equal to Pls (X = "Success") = 7.49e − 3. It is worth noting that values of *Bel (Attack = "Success")* and *Pls (Attack = "Success")* are

Table 1
CBT of "Docking" as an AND gate.

	Patrol	Success			Failure			Success, Failure		
	Docking_Barriers	Success	Failure	Success, Failure	Success	Failure	Success, Failure	Success	Failure	Success, Failure
Docking	Success	1	0	0	0	0	0	0	0	0
	Failure	0	1	0	1	1	1	0	1	0
	Success, Failure	0	0	1	0	0	0	1	0	1

Table 2
CBT of "Attack" as an OR node.

	Attack_via_Water	Success			Failure			Success, Failure		
	Attack_via_Ground	Success	Failure	Success, Failure	Success	Failure	Success, Failure	Success	Failure	Success, Failure
Attack	Success	1	1	1	1	0	0	1	0	0
	Failure	0	0	0	0	1	0	0	0	0
	Success, Failure	0	0	0	0	0	1	0	1	1

Table 3
CBT for Bel_A_Success. This node reports the value of Bel(Attack = "Success") as an OR gate.

Bel_A_Success	Attack = "Success"	Attack = "Failure"	Attack = "Success, Failure"
Believe (Attack = "Success")	1	0	0
Not Believe (Attack = "Success")	0	1	1

Table 4
CBT for Pls_A_Success. This node reports the value of Pls(Attack = "Success") as an OR gate.

Pls_A_Success	Attack = "Success"	Attack = "Failure"	Attack = "Success, Failure"
Plausible (Attack = "Success")	1	0	1
Not Plausible (Attack = "Success")	0	1	0

Table 5
Basic Belief Assignment for the primary nodes.

Variable X	M (X = "Success")	M (X = "Failure")	M (X = "Success, Failure")
Main_Gate	0.1	0.8	0.1
First_Fence	0.3	0.6	0.1
Patrol	0.2	0.7	0.1
Docking_Barriers	0.2	0.7	0.1
CCTV	0.2	0.7	0.1
Second_Fence	0.3	0.6	0.1
IED	0.7	0.2	0.1
Regress	0.4	0.5	0.1

Table 6
Results obtained from the EN using GeNIe.

Variable X	Bel (X = "Success")	Pls (X = "Success")
Attack_via_Ground	1.24e−3	7.49e−3
Attack_via_Water	6.72e−4	4.32e−3
Attack	1.67e−3	9.56e−3

the highest among the values reported in Table 6. This is reasonable because the variable "Attack" is intended to represent a generic attack, which can be either via ground, or via water.

Specifying the CBT of this variable as an OR node implies the possibility that both scenarios can occur at the same time. This can happen, for example, when a group of adversaries decides to attack from different paths to have higher possibilities to achieve success. If a single adversary capable of carrying out only one kind of attack is considered (e.g., the attacker is a lone-wolf), the CBT of "Attack" node should be defined as a XOR (i.e., Exclusive OR) gate, to exclude the possibility that both attack paths can be followed at the same time. The XOR logic gate produces a true output only if one, and only one of the inputs is true (Simpson, 1987). In practice, the OR-CBT reported in Table 2 should be substituted with the XOR-CBT presented in Table 7.

3.3. Converting Attack trees to Credal network

Differently from EN, CN can be used both to conduct forward analysis and to update probabilities. CNs are based on a generalization of Bayes' theorem, as presented in Section 2. In order to make a direct comparison between the two methodologies, a CN equivalent to the EN presented in the previous subsection has been developed using the software JavaBayes (Cozman) in Fig. 10.

As shown in the figure, the names of the variables are the same used in the previous subsection. In the CN, however, the variables are binary, and the possibility space of each of them comprises only the states "Success" and "Failure". CSs describing binary variables are probability intervals, thus, in order to assign probabilities to primary nodes, the values presented in Table 5 were converted to probability bounds using *Bel* and *Pls* functions, and conservatively considering the widest interval using Eq. (13):

$$[\underline{P}(X = x), \overline{P}(X = x)] = [Bel(X = x), Pls(X = x)] \quad (13)$$

where X is a variable, and x is one of the allowed states of the variable X. Prior interval-valued probabilities used in the CN model are reported in Table 8. As shown in Section 2, the methodology proposed by Walley (1991) to translate language in probability bounds may also be adopted, since it represents a simple framework converting verbal judgments into probabilistic constraints, which may be of great interest due to the fact that SVA heavily relies on the value of experts'

Table 7
CBT of "Attack" as a XOR gate.

	Attack_via_Water	Success			Failure			Success, Failure		
	Attack_via_Ground	Success	Failure	Success, Failure	Success	Failure	Success, Failure	Success	Failure	Success, Failure
Attack	Success	0	1	0	1	0	0	0	0	0
	Failure	1	0	0	0	1	0	0	0	0
	Success, Failure	0	0	1	0	0	1	1	1	1

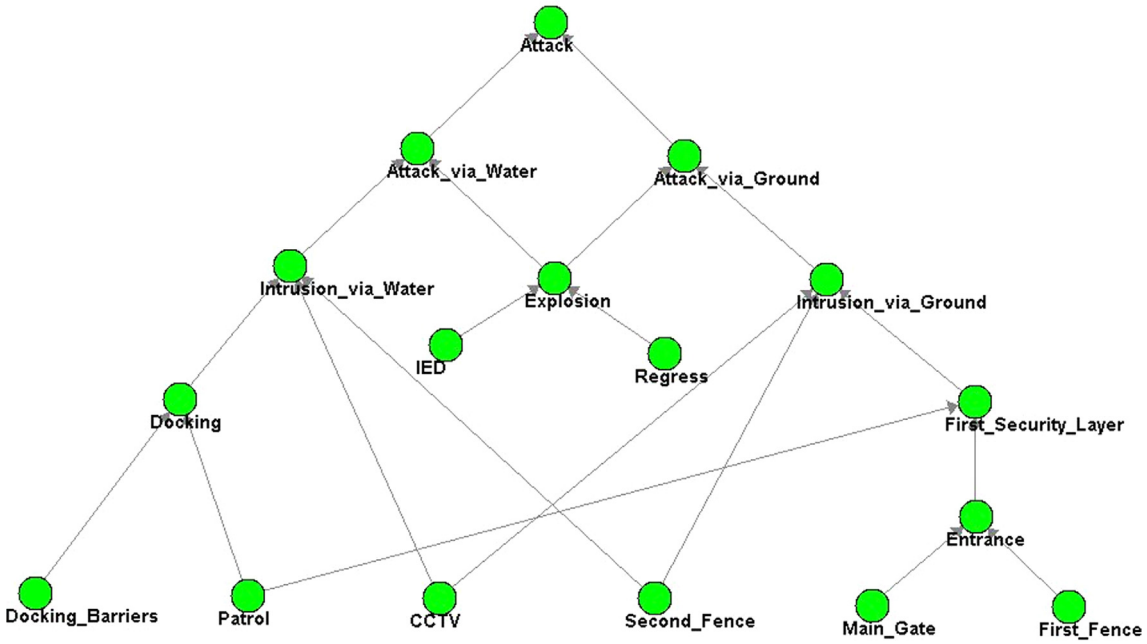

Fig. 10. CN of the attack scenarios.

Table 8
Probability intervals assigned to primary nodes in the CN.

Variable	P (X = "Success")	P (X = "Failure")
Main_Gate	[0.1, 0.2]	[0.8, 0.9]
First_Fence	[0.3, 0.4]	[0.6, 0.7]
Patrol	[0.2, 0.3]	[0.7, 0.8]
Docking_Barriers	[0.2, 0.3]	[0.7, 0.8]
CCTV	[0.2, 0.3]	[0.7, 0.8]
Second_Fence	[0.3, 0.4]	[0.6, 0.7]
IED	[0.7, 0.8]	[0.2, 0.3]
Regress	[0.4, 0.5]	[0.5, 0.6]

Table 9
CPT for "Docking" as an AND gate.

	Water_Patrol	Success		Failure	
	Docking_Barriers	Success	Failure	Success	Failure
Docking	Success	1	0	0	0
	Failure	0	1	1	1

Table 10
CPT for "Entrance" as an OR gate.

	Main_Gate	Success		Failure	
	First_Fence	Success	Failure	Success	Failure
Entrance	Success	1	1	1	0
	Failure	0	0	0	1

Table 11
Results obtained querying the CN through Javabayes.

Variable X	*JavaBayes. P (X = "Success")*
Attack_via_Ground	[1.24e−3, 7.49e−3]
Attack_via_Water	[6.73e−4, 4.32e−3]
Attack	[1.67e−3, 9.56e−3]

knowledge. However, in this case study the aim is to compare the two methodologies, so the same input data is used.

Conditional probability tables for AND and OR gates have been realized following the same approach adopted in BN (Bobbio et al., 2001). For example, Table 9 reports the CPT of an AND gate (e.g., "Docking") with respect to its parents.

Table 10 reports the CPT for an OR gate (e.g., "Entrance") with respect to its parents.

In order to perform forward analysis, the package JavaBayes was used. This software tool was originally intended to compute robustness analysis of BNs, but can incorporate the CPTs as CSs. The software is able to compute only CNs according to strong extension.

The results obtained querying the CN are reported in Table 11.

As can be seen from Tables 6 and 11, the results obtained through the two methodologies are the same.

Mapping ATs to CN allows updating probabilities given new evidence. This, however, is not allowed with DST. Since computing PPSs' posterior probability after a security event may be of great interest to make diagnosis of security system, JavaBayes was used for updating probabilities. The updating algorithm in CN usually produces wide posterior intervals (Seidenfeld and Wasserman, 1993), thus the results are not always informative though the methodology is more robust than BN, due to a lower amount of information needed to start the calculation (i.e., prior probabilities do not need to be point values, and a range of values is sufficient).

Results obtained querying the model, given new evidence, are reported in Table 12. Two tests have been made: (i) setting the evidence "Attack = Success" and computing the updated reliability of the PPSs, and (ii) setting the evidence "Attack = Failure" and again computing the updated reliability of PPSs.

From Table 12, it is evident that uncertainty among posteriors is generally higher than among priors. This is due to the expansion of posteriors expected in CN updating (Seidenfeld and Wasserman, 1993). Actually, setting Attack = "Success" may lead to unitary posterior probability of some variables. For example, P (CCTV = "Success" | Attack = "Success") = 1.0, indicating that in case of a successful attack the CCTV must have successfully disabled or eluded already by the attackers. In the next section, this problem is outlined with an example, and relaxation of the deterministic AND/OR gates is proposed in order to compute CN.

Table 12
Posterior Intervals of primary nodes obtained through JavaBayes.

X = "x"	P (X = "x" | Attack = "Success")	P (X = "x" | Attack = "Failure")
Main_Gate = "Success"	[1.61e−1, 3.62e−1]	[9.93e−2, 2.00e−1]
First_Fence = "Success"	[4.93e−1, 7.04e−1]	[2.98e−1, 4.00e−1]
Patrol = "Success"	1	[1.95e−1, 2.98e−1]
CCTV = "Success"	1	[1.95e−1, 2.98e−1]
Second_Fence = "Success"	1	[2.95e−1, 3.99e−1]
IED = "Success"	1	[6.97e−1, 8.00e−1]
Regress = "Success"	1	[3.95e−1, 4.99e−1]
Docking_Barriers = "Success"	[3.25e−1, 5.37e−1]	[1.99e−1, 3.00e−1]

Table 13
Posterior Intervals of primary nodes obtained through JavaBayes, and through GL2U.

X = "x"	JavaBayes. P (X = "x" | Attack = "Success")	GL2U. P (X = "x" | Attack = "Success")
Main_Gate = "Success"	[1.61e−1, 3.62e−1]	[9.70e−2, 1.99e−1]
First_Fence = "Success"	[4.93e−1, 7.04e−1]	[2.90e−1, 3.98e−1]
Patrol = "Success"	1	[1.77e−1, 2.93e−1]
CCTV = "Success"	1	[1.22e−1, 2.42e−1]
Second_Fence = "Success"	1	[2.11e−1, 3.50e−1]
IED = "Success"	1	[7.26e−1, 9.99e−1]
Regress = "Success"	1	[4.52e−1, 9.97e−1]
Docking_Barriers = "Success"	[3.25e−1, 5.37e−1]	[1.91e−1, 2.98e−1]

Table 14
CPT for relaxed AND gate of "Docking".

	Patrol	Success		Failure	
	Docking_Barriers	Success	Failure	Success	Failure
Docking	Success	0.99	0.01	0.01	0.01
	Failure	0.01	0.99	0.99	0.99

Table 15
CPT for relaxed OR gate of "Entrance".

	Main_Gate	Success		Failure	
	First_Fence	Success	Failure	Success	Failure
Entrance	Success	0.99	0.99	0.99	0.01
	Failure	0.01	0.01	0.01	0.99

Table 16
Results obtained through JavaBayes and GL2U, with relaxation assumption to the logic gates.

Variable X	*JavaBayes: P (X = "Success")*	*GL2U: P (X = "Success")*
Attack_via_Ground	[1.41e−2, 2.15e−2]	[1.40e−2, 2.10e−2]
Attack_via_Water	[1.36e−2, 1.85e−2]	[1.40e−2, 1.80e−2]
Attack	[3.67e−2, 4.67e−2]	[3.70e−2, 4.90e−2]

4. Discussion

4.1. Comparison of results

As reported above, the results obtained in forward analysis through DST and CN are coincident. It was also shown that the DST belief propagation approach is applicable to multiconnected networks. This result is worthwhile, since previous applications have been limited to a single FT, resulting in a polytree-shaped EN (Simon et al., 2008; Simon and Weber, 2009). In the present study, however, the methodology was applied to merge multiple attack trees in a single multiconnected EN. The results were compared with those obtained through JavaBayes, which in forward analysis adopts the same Bayesian propagation algorithms employed in GeNIe. The results are in sufficient agreement, thus suggesting that it is possible to use a standard Bayesian propagation algorithm, embedded in commercial packages, to merge trees and analyse multi-connected ENs.

4.2. Evidential network vs. Credal network

EN allows to explicitly depict the propagation of epistemic uncertainty thanks to the presence of the epistemic state. Actually, variables are binary, but BBA is carried out for the four elements of the power set of the states. For example, for a variable X whose possible states are x_1 and x_2, the power set is $\{\varnothing,x_1,x_2,\{x_1,x_2\}\}$, where the last element is the epistemic state. According to Eq. (8) no belief mass was assigned to the first element (e.g. $X = \varnothing$), that was thus neglected in the EN. A belief mass was assigned to the other three elements. Forward belief propagation was computed as if the network was a BN comprising only ternary variables, even if the epistemic states represent the amount of uncertainty affecting each node rather than variables' states. Keeping track of epistemic uncertainty through the epistemic state may be of great relevance to understand where the lack of information is more relevant.

Despite the advantages of EN in uncertainty propagation modelling, no specific software is available; as such, algorithms originally intended for BN were used for this purpose (Simon et al., 2008; Simon and Weber, 2009).

EN should not be considered as a probabilistic tool, since the DST is based on belief masses rather than probabilities. Indeed, probability bounds obtained through CN with values of *Bel* and *Pls* functions were compared in the present study, instead of belief masses. Thus, in DST, *Bel* and *Pls* can be interpreted as tools to reconstruct probabilities from experts' basic belief assignment. The non-probabilistic nature of DST leads to one of the major shortcomings of EN with respect to CN, that is, EN cannot perform probability updating. Since DST is a more general framework than probability theory, belief mixing rules would be needed to take into account new information, and these rules are not embedded in commercial Bayesian software.

As previously outlined, only a few packages to deal with DST are available (IPP Toolbox; Smets, 2004) though they are not easy to use, and are usually designed to compute single Dempster-Shafer structures rather than ENs. In the literature, there is a huge criticism against mixing rules (Zadeh, 1979; Tchamova and Dezert, 2012). On the contrary, CN can take the advantage of a generalization of the Bayes' Theorem, and thus algorithms to efficiently compute it are embedded in the available packages.

4.3. Different tools may provide different results in Credal network

The main advantage of CN over EN is the possibility of updating the results given new evidence. The updating procedure can be carried out through algorithms embedded in tools specifically developed for Credal Sets Theory, mainly experimental and open-source, as GL2U (Sun, 2008) and JavaBayes (Cozman). Despite the fact that in the previous section only the results of JavaBayes are reported for forward analysis, applicability of GL2U was also examined. GL2U is hard to set up and not intuitive to use, leading to difficulties in developing the network, and in interpretation of the outcomes. GL2U is designed to compute only purely probabilistic networks, and the necessity to map AND Gates can create some problems during updating. For example, in Table 13 posterior probabilities obtained through GL2U, given a successful attack, are compared with those obtained through JavaBayes, presented in Table 12.

As evident from the table, the results are not in agreement. Considering a successful attack, results obtained through GL2U for the variables "CCTV" and "Second_Fence" are not consistent. Since these

Table 17
Attempt to update Dempster-Shafer structures of primary nodes, given the evidence of "Attack = Failure".

X = "x"	Javabayes	M (X = "x")	M (X = "−x")	M (X = "x, −x")
Main_Gate = "Success"	[9.93e−2, 2.00e−1]	9.93e−2	8.00e−1	1.01e−1
First_Fence = "Success"	[2.98e−1, 4.00e−1]	2.98e−1	6.00e−1	1.02e−1
Patrol = "Success"	[1.95e−1, 2.98e−1]	1.95e−1	7.02e−1	1.03e−1
CCTV = "Success"	[1.95e−1, 2.98e−1]	1.95e−1	7.02e−1	1.03e−1
Second_Fence = "Success"	[2.95e−1, 3.99e−1]	2.95e−1	6.01e−1	1.04e−1
IED = "Success"	[6.97e−1, 8.00e−1]	6.97e−1	2.00e−1	1.03e−1
Regress = "Success"	[3.95e−1, 4.99e−1]	3.95e−1	5.01e−1	1.04e−1
Docking_Barriers = "Success"	[1.99e−1, 3.00e−1]	1.99e−1	7.00e−1	1.01e−1

Table 18
Attempt to update Dempster-Shafer structures of primary nodes, given the evidence of "Attack" = "Success".

X = "x"	Javabayes	M (X = "x")	M (X = "−x")	M (X = "x, −x")
Main_Gate = "Success"	[1.61e−1, 3.62e−1]	1.61e−1	6.28e−1	2.11e−1
First_Fence = "Success"	[4.93e−1, 7.04e−1]	4.93e−1	2.96e−1	2.11e−1
Patrol = "Success"	1	1	0	0
CCTV = "Success"	1	1	0	0
Second_Fence = "Success"	1	1	0	0
IED = "Success"	1	1	0	0
Regress = "Success"	1	1	0	0
Docking_Barriers = "Success"	[3.25e−1, 5.37e−1]	3.25e−1	4.63e−1	2.12e−1

primary nodes are connected with AND gates, the states of these variables have necessarily to be "Success" given a successful attack. For example, the attackers' successful evasion of the CCTV system, according to the ATs, is a necessary step to have a successful intrusion. The same rationale can be followed observing the values obtained for "IED", and "Regress". As stated at in Section 3, the attackers consider an attack successful if and only if they manage to regress from the plant. Thus, given a successful attack, the value of "Regress" = "Success" must be unitary, which is not the case in GL2U.

This inconsistency is due to the algorithm employed by GL2U. Indeed, the implemented code is not able to compute updating in presence of variables whose CPTs contains only zeros and ones. For more information about this issue, the reader can refer to literature concerning zero-conditioning events (Corani et al., 2012; Walley, 1991). Thus, GL2U is applicable for backward analysis only in case of purely probabilistic networks, that is, networks not containing any deterministic relation (e.g., it cannot be applied to compute diagnostic analysis in case of presence of AND and OR logics).

4.4. Relaxing AND and OR gates in Credal network

A possible solution to the issue of zero-conditioning could be to relax deterministic AND and OR gates, mapping the ATs in a purely probabilistic CN. An approach that may be applied is to adopt an imprecise noisy-OR gate, as presented by Antonucci (2011). However, in Antonucci's work only an extension of the OR gate is presented, while an extension for the AND gate is not outlined, and the definition of the imprecise noisy-OR gate requires the specification of additional assumptions to define the CS.

Noisy OR-Gate has been defined for the first time by Pearl (1988), as a disjunctive operator adoptable to reduce the number of parameters needed to set up BN. An interesting possibility can be to adopt the framework proposed by Henrion (1989), that is, to model hidden causes together in a leak probability. The leak parameter may be used to express the lack of possibility to experimentally define all the possible causes leading to some effect, and for this reason it may be suitable to account for epistemic uncertainty about dependencies. Likewise, the noisy AND-gate is an operator to account for multiple necessary conditions to achieve a result. The reader may refer to Galán and Díez (2000) for more information about noisy AND-gate, and leaky noisy AND-gate.

The approach followed in our analysis is to change the CPTs, replacing "1" values with "near-to-1" values, e.g., 0.99. Tables 14 and 15 respectively show the relaxed CPTs adopted for AND and OR gates of "Docking", and "Entrance" nodes.

Adopting this relaxed framework, the CN comprises only probabilistic nodes, and the algorithms can deal with the computation. Results obtained in forward analysis through the two packages are compared in Table 16.

According to Table 16, results obtained in forward analysis using the two packages are in agreement. It is remarkable that the level of vulnerability, i.e., the probability of a successful attack, adopting relaxed gates according to Tables 14 and 15 are more severe than those considering deterministic nodes. For example, the probability interval that "Attack" = "Success" was [1.67e−3, 9.56e−3], considering deterministic OR and AND gates (i.e., Table 11), while relaxing the gates it turns out to be [3.67e−2, 4.67e−2]. Considering that ATs are tools intended to represent situations of high uncertainty (i.e., the analyst may not be able to outline perfectly the options available to the attackers), and comparing data of Tables 11 and 16, it can be concluded that the usage of deterministic gates may be not suitable for all the applications. Therefore, given the high level of uncertainty affecting security vulnerability assessment, it may be better to soften the operators, e.g., via noisy gates.

4.5. Attempt to update EN through the equivalent CN

The main advantage of CN over EN is the possibility of updating the information, given new evidence. In case a backward analysis is needed, EN is not applicable. An idea can be to construct a BN considering the mean value of [Bel (X = x), Pls (X = x)], and then to update the probabilities of the primary nodes given new evidence. Zhang and Thai (2015) adopted this approach to simplify a CN in a case of reliability analysis in maritime engineering. However, there is no way to reconstruct probability intervals only from these updated middle points. Thus, this approach leads to loss of information concerning the epistemic uncertainty among variables.

This is the reason why computing updating through an equivalent CN was considered in the present study. After finding posterior intervals corresponding to [Bel (X = x | evidence), Pls (X = x | evidence)], the Dempster-Shafer structures (i.e., the distribution of belief masses) could be reconstructed through Möbius Transformation (Smets, 2002), and

Table 19
Comparison of BN, EN, and CN.

Network	Main Features	Available tools	Suggested application	Drawbacks
Bayesian Network (BN)	Flexible graphical probabilistic tool for both prognostic and diagnostic analyses, thanks to Bayes' Theorem (Pearl, 1988; Charniak, 1991). BN is relatively easy to construct. Standardized algorithms are available for inference	Well-known algorithms and software are available (Pearl, 1988). GeNIe is a good example of such widespread software tools (GeNle Modeler)	Mapping of complex systems, if values of conditional probabilities are slightly affected by epistemic uncertainty (i.e., if quantification of probabilities as point values is not an issue) (Bobbio et al., 2001; Khakzad et al., 2011; Khakzad et al., 2013; Van Staalduinen et al., 2017)	Marginal and conditional probabilities of events have to be defined as point values (i.e., second-order uncertainty cannot be accounted for)
Credal Network (CN)	Graphically equivalent to BN, but marginal and conditional probabilities are replaced with closed convex sets of probability (i.e., credal sets) (Corani et al., 2012; Walley, 1991; Cozman, 2000; Piatti et al., 2010; Avis and Fukuda, 2000; Avis and Fukuda, 1992; Walley, 1996; Antonucci and Zaffalon, 2008; Couso et al., 1999; Antonucci, 2008; Cozman; Cano et al., July 1994; Yi, 2008; Ide, 2004; Antonucci et al., 2004; Antonucci et al., 2013). Credal Sets can be directly reconstructed from experts' verbal judgments (Walley, 1991). An extension of Bayes' Theorem for Credal Sets is available, allowing diagnostic (backward) analysis (Cozman, 2000)	Algorithms for CN computation are inefficient and poorly standardized (Corani et al., 2012). Nevertheless, open-source codes such as JavaBayes (Cozman) and GL2U (Yi, 2008) have been developed for this purpose	CN should be employed for mapping complex systems in case probabilities cannot be estimated as point values: specifying dependences as sets of probability leads to more robust results. It is possible to employ CN for backward analysis in systems where marginal probabilities are affected by uncertainty	CN is a more complex tool than BN. The developed software is not easy-to-use, and may lead to unrealistic results in backward analysis (i.e., the special case of zero-conditioning events mentioned in Section 4.3) CN is a purely probabilistic tool, which should not be employed in mapping deterministic logics (e.g. AND gates), as mentioned in Section 4.4
Evidential Network (EN)	Ideal for explicit quantification of the amount of belief affected by epistemic uncertainty; ideal for explicit propagation of uncertainty within system's elements	A few software packages has been designed specifically for EN; however, standardized BN software can be used for EN computation in forward analysis. The specification of uncertainty through the epistemic state is more direct and explicit than in CN	EN should be employed for mapping complex systems where epistemic uncertainty is relevant if keeping track of its propagation among variables is explicitly required. It may have the same applications of CN, but since EN is more intuitive to be defined and easier to be computed, it is specifically suggested if backward analysis is not needed	Since EN is not a probabilistic tool, and since mixing rules for "belief updating" may lead to unrealistic results, the updating procedure is not possible with EN, limiting its application to cases where only forward analysis is required. In Section 4.5 an approximate updating procedure for EN (through the equivalent CN) has been proposed. Nevertheless, the authors suggest to directly employ CN for diagnostic analysis

updated results were calculated. For example, updated outcomes for primary variables given "Attack" = "Failure" are presented in Table 17 in term of belief masses. For the sake of clarity, in each row the interval-valued posteriors of primary nodes obtained through JavaBayes are reported along with the belief masses of the corresponding Dempster-Shafer structures.

In Table 17, "x" refers to the first outcome, i.e., "Success", whereas "$-x$" refers to the second outcome, i.e., "Failure". The last column contains the belief mass associated to the epistemic state, labelled with "$x,\ -x$".

For example, the posterior probability interval found through JavaBayes for Main_Gate = "Success" is [9.93e − 2, 2.00e − 1], resulting in a belief mass of 9.93e − 2 for "Success", a belief mass of 8.00e − 1 for "Failure", and a belief mass of 1.01e − 1 for epistemic uncertainty.

It is worth noting that the mass left to the uncertainty has slightly grown for each variable: this was actually expected, because the updating procedure in CN enhances the size of CSs (i.e., in this case the width of the probability interval) (Seidenfeld and Wasserman, 1993).

In Table 18 the updated outcomes are reported for primary nodes, given a successful attack.

It is worth noting that in this case, five variables have no uncertainty, because the reported outcomes (e.g., CCTV = "Success") are necessary conditions for a successful attack, given the presence of AND gates. The other three variables, on the contrary, are affected by high uncertainty.

4.6. Summary of results

Sections 4.1–4.5 provide some comments based on the application of the EN and CN to the case study in Section 3, with an emphasis on the applicability of the tools, and their relative advantages and drawbacks. The results have been summarized in Table 19.

5. Conclusions

In the present study, two methodologies to assess epistemic uncertainty and imprecision have been presented, together with a comparison of their respective characteristics: (i) DST-based network, also known as Evidential network (EN), and (ii) Credal network (CN), a generalization of Bayesian network. Evidential network constitutes an intuitive framework for experts to assign belief masses in case of epistemic uncertainty, or necessity to use expertise. It is not intended to be a probabilistic tool, and the network propagates degrees of belief rather than probabilities. On the contrary, Credal network is a probabilistic tool, very similar to Bayesian network. Conditional probabilities are replaced with closed convex sets of probability masses called Credal sets (CSs). Tools to directly convert constraints from experts' judgments to Credal sets have been found, and may be used to directly reconstruct a probabilistic framework from verbal comparative judgments.

In order to compare the methodologies, two ATs have been mapped and then analysed through Evidential network and Credal network to compute the level of vulnerability of the security system, given two scenarios. Evidential network is easier to implement because it can be built up using commercial Bayesian packages like GeNIe (GeNIe Modeler); Credal network can be computed only employing experimental and open-source codes, like JavaBayes (Cozman). The two methodologies produced the same outcomes, so they are mainly equivalent. Actually, the meaning of the networks is slightly different. Evidential network is a directed acyclic graph propagating belief masses, while Credal network propagates probabilities.

Regarding the usability of the methodologies, Evidential network seems more intuitive. Indeed, the presence of the epistemic state may be useful to experts to assign directly the amount of information they are not sure about. Credal network requires the prior definition of Credal sets, which is relatively easy for binary variables, but may become time demanding for multinomial variables. Furthermore, the explicit propagation of epistemic uncertainty in the Evidential network through the presence of the epistemic state allows to keep track of the effect of imprecision on the computation of results, while in Credal network the vagueness has to be estimated from the extent of Credal sets, the width of probability intervals in case of binary variables.

Although Credal network can be employed to conduct backward diagnostic analysis, updating leads to the expansion of the intervals' width producing less informative outcomes. Since Credal network and Evidential network have been found to produce equivalent results in forward analysis, we propose to approximately update Evidential network, considering its equivalent updated Credal network. From posterior probability intervals it is then possible to reconstruct updated values of Evidential network, to be used to revise experts' belief assignments.

In summary, in case only forward analysis is needed, Dempster-Shafer Theory can be used to map Attack trees into Evidential network because the creation of the network is intuitive and allows to explicitly keep track of propagation of uncertainty. Experts may easily asses a priori beliefs about variables, and standardized commercial software can be employed. If backward analysis is needed to update knowledge given new evidence, Credal network represents a theoretically more solid approach.

References

Antonucci, A., Salvetti, A., Zaffalon, M., 2004. Assessing debris flow hazard by credal nets. In: Lopez-Diaz et al. (Eds.), Soft Methodology and Random Information Systems. Springer-Verlag, Berling, Heidelberg.

Antonucci, A., Huber, D., Zaffalon, M., Luginbuhl, P., Chapman, I., Ladouceur, R., 2013. CREDO: a military decision-support system based on credal networks. In: 16th International Conference on Information Fusion (FUSION 2013), Istanbul, Turkey.

Antonucci, A., Zaffalon, M., 2008. Decision-Theoretic specification of credal networks: a unified language for uncertain modelling with sets of Bayesian networks. Int. J. Approx. Reason. 49 (2), 345–361.

Antonucci, A., 2008. Imprecise Probabilistic Graphical Models: Equivalent Representations, Inference Algorithms and Applications, Doctoral Dissertation, Faculty of Informatics, University of Lugano.

Antonucci, A., 2011. The imprecise noisy-OR gate. In: 14th International Conference on Information Fusion.

API (American Petroleum Institute), 2003, API RP-70: Security for Offshore Oil and Natural Gas Operations.

Argenti, F., Landucci, G., Reniers, G., 2016. Probabilistic vulnerability assessment of chemical plants subjected to external acts of inference. Chem. Eng. Trans. 48, 691–696.

Avis, D., Fukuda, K., 2000. A revised implementation of the reverse search vertex enumeration algorithm. In: Kalai, G., Ziegler, G. (Eds.), Polytopes-Combinatorics and Computation, pp. 177–197.

Avis, D., Fukuda, K., 1992. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedral. Discrete Comput. Geometry 8, 295–313.

Bajpai, S., Gupta, J.P., 2005. Site security for chemical process industries. J. Loss Prev. Process Ind. 18, 301–309.

Baybutt, P., Ready, V., 2003. Strategies for protecting process plants against terrorism, sabotage and other criminal acts. Homeland Defence J. 2, 1.

Benavoli, A., Ristic, B., Farina, A., Oxenham, M., Chisci, L., 2009. An application of evidential networks to threat assessment. IEEE Trans. Aerospace Electron Syst., 45(2), 620–639 (pp. 713–722).

Bhashyam, S.S., Montibeller, G., 2016. In the opponent's shoes: increasing the behavioral validity of attackers' judgments in counterterrorism models. Risk Anal. 36 (4), 666–680.

Bobbio, A., Portinale, L., Minichino, M., Ciancamerla, E., 2001. Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. Reliab. Eng. Syst. Saf. 71, 249–260.

Brooke, P.J., Paige, R.F., 2003. Fault trees for security system design and analysis. Comput. Security 22 (3), 256–264.

Cano, A., Cano, E., Moral, S., 1994. Convex sets of probabilities propagation by simulated annealing. In: Goos, G., Hartmanis, J., van Leeuwen, J. (Eds.), Proceeding International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Paris, France, July 1994.

CCPS (Centre for Chemical Process Safety), 2003. Guidelines for Analyzing and Managing the Security Vulnerabilities of Fixed Chemical Sites. American Institute of Chemical Engineers. Wiley, New York, 13:978-0816908776.

Charniak, E., 1991. Bayesian Networks without Tears. AAAI, AI Magazine 12 (4), 50–63.

Cheng, Y.L., 2000. Uncertainty in fault tree analysis. Tamkang J. Sci. Eng. 3 (1), 23–29.

Corani, G., Antonucci, A., Zaffalon, M., 2012. Bayesian networks with imprecise probabilities: theory and application to classification. In: Data Mining: Found. & Intell. Paradigms, ISRL, vol. 23, pp. 49–93.

Couso, I., Moral, S., Walley, P., 1999. Examples of independence for imprecise probabilities. In: 1st International Symposium on Imprecise Probabilities and Their

Applications, Ghent, Belgium, 29 June - 2 July 1999.
Cozman, F.G., 2000. Credal networks. Artif. Intell. 120, 199–233.
Cozman F. G. JavaBayes – user manual. < http://www.cs.cmu.edu/~javabayes/ > .
Dempster, A.P., 1967. Upper and lower probabilities induced by multi-valued mapping. Ann. Math. Stat. 38, 325–339.
Fakhravar, D., Khakzad, N., Reniers, G., Cozzani, V., 2017. Security vulnerability assessment of gas pipelines using Discrete-time Bayesian network. Process Saf. Environ. Prot. 111, 714–725.
Galán, S.F., Díez, F.J., 2000. Modelling dynamic causal interactions with Bayesian networks: temporal noisy gates. In: Working Notes of CaNew'2000, the 2nd International Workshop on Causal Networks, ECAI-2000, Berlin (Germany), pp. 1–5.
GeNIe Modeler, BayesFusion, LCC, < http://www.bayesfusion.com/ > .
Gribaudo, M., Iacono, M., Marrone, S., 2015. Exploiting Bayesian networks for the analysis of combined attack trees. Electron. Notes Theor. Comput. Sci. 310, 91–111.
Guth, M., 1991. A probabilistic foundation for vagueness and imprecision in fault tree analysis. IEEE Trans. Reliab. 40 (5), 563–570.
Henrion, M., 1989. Some practical issues in constructing belief networks. In: Kanal, L.N., Levitt, T.S., Lemmer, J.F. (Eds.), Uncertainty in Artificial Intelligence 3. Elsevier Science Publishers, pp. 161–173.
Ide, J.S., Cozman, F.G., 2004. IPE and L2U: Approximate algorithms for credal networks, Second Starting AI Researcher Symposium (STAIRS). IOS Press, pp. 118–127.
Ingoldsby, T.R., 2013, Attack Tree-based Threat Risk Analysis, Amenaza Technologies Limited.
IPP Toolbox 1.0, GNU Public Licence. < https://www.uni-due.de/informationslogistik/ipptoolbox.php > .
Kay, Rakowsky U.K., 2007. Fundamentals of the Dempster-Shafer theory and its Applications to Systems Safety and Reliability Modelling, RTA 3-4, December, Special Issue.
Khakzad, N., Khan, F., Amyotte, P., 2011. Safety analysis in process facilities: comparison of fault tree and Bayesian network approaches. Reliab. Eng. Syst. Saf. 96, 925–932.
Khakzad, N., Khan, F., Amyotte, P., 2013. Dynamic safety analysis of process systems by mapping bow-tie into Bayesian network. Process Saf. Environ. Prot. 91, 46–53.
Limbourg, P., Savic, R., Petersen, J., Kochs, H.-D., 2007. Fault tree analysis in an early design stage using the Dempster-Shafer theory of evidence. In: Aven & Vinnem (Eds.), Risk, Reliability and Societal Safety.
Nai, Fovino I., Masera, M., De Clan, A., 2009. Integrating cyber attacks within fault trees. Reliab. Eng. Syst. Saf. 94, 1394–1402.
Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco, CA.
Piatti, A., Antonucci, A., Zaffalon, M., 2010. Building knowledge-based systems by credal networks: a tutorial. In: Baswell, A.R. (Ed.), Advances in Mathematics Research, vol. 11. Nova Science Publishers, New York.
Schneier B., 1999, Attack Trees, Dr. Dobb's Journal, December.
Schneier, B., 2000. 1st ed., Secrets & Lies: Digital Security in a Networked World. John Wiley & Sons, Inc., New York, NY, USA.
Seidenfeld, T., Wasserman, L., 1993. Dilation for sets of probability. Ann. Stat. 21 (3), 1139–1154.
Shafer, G., 1976. A Mathematical Theory of Evidence. Princeton University Press, Princeton, NJ.
Simon, C., Weber, P., Evsukoff, A., 2008. Bayesian networks inference algorithm to implement Dempster Shafer theory in reliability analysis. Reliab. Eng. Syst. Saf. 93, 950–963.
Simon, C., Weber, P., 2009. Evidential networks for reliability analysis and performance evaluation of systems with imprecise knowledge. IEEE Trans. Reliab. 58 (1), 69–87.
Simpson, R.E., 1987. The Exclusive OR (XOR) Gate, in Introductory Electronics for Scientists and Engineers, 2nd ed. Allyn and Bacon, Boston, MA, pp. 550–554 (§12. 5.6).
Smets, P., 2002. The application of the matrix calculus to belief functions. Int. J. Approx. Reason. 31, 1–30.
Smets Ph., 2004. TBMLAB. < http://iridia.ulb.ac.be/~psmets/#G >
SUN Yi, 2008. The GL2U Package, < http://people.idsia.ch/~sun/gl2u_brief.pdf>.
Tchamova, A., Dezert, J., 2012. On the Behavior of Dempster's Rule of Combination and the Foundations of Dempster-Shafer Theory, IEEE IS'2012, Sofia, Bulgaria, Sept. 6–8.
US Nuclear Regulatory Commission, 1981 (January), Fault Tree Handbook, NUREG-0492.
Van Staalduinen, M.A., Khan, F., Gadag, V., Reniers, G., 2017. Functional quantitative security risk analysis (QSRA) to assist in protecting critical process infrastructures. Reliab. Eng. Syst. Saf. 157, 23–34.
Walley, P., 1991. Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, New York, NY, USA.
Walley, P., 1996. Measures of uncertainty in expert systems. Artif. Intell. 83, 1–58.
Zadeh, L., 1979. On the validity of Dempster's rule of combination, Memo M79/24. University of California, Berkeley, US.
Zhang, G., Thai, V.V., 2015. Maritime accidents risk prediction based on Bayesian Network with interval probabilities. Safety Reliab. Complex Eng. Syst. 2018–2024.