A Ranking Metric for Event Impact in Safety-Security Co-Analysis

Rosan Maas
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
r.f.maas@student.utwente.nl

ABSTRACT

In an ever-evolving world, safety and security are of great importance. Many ways to analyse the risks of safety and security have emerged in the past years. Yet often these methods are either non-quantitative or fail to take into account the interdependencies between the two. We propose a metric to rank events based on their impact on both safety and security to add a quantitative method for co-analysing the two. We use a case study to apply this metric and see whether or not this metric is a proper way of ranking the impact of events on both safety and security. We have modelled this case using a Bayesian network. We have found that though the metric gives a broad overview over the impact of events and their ranking, it is not something that can be used separately from analysis about dependencies and network depth.

Keywords

Safety, Security, Modelling, Dependencies, Bayesian Networks, Ranking, Autonomous Surface Vessel

1. INTRODUCTION

Operational risks always have regard to safety, security or both. The difference between safety and security is the intent [13]. Where safety concerns accidental failures, for example, a fire or a rope snapping in two because of wear, security concerns malicious intent, like cutting the rope in two for example. Of course this can also interact, think about someone partially cutting a rope so it will snap in two faster, in that case the attacker makes use of a safety risk.

As we gain more and more intelligent systems, systems become more complex and risks become less clear. The line between safety and security is often blurred [19].

In recent years, research into safety, security, the related risks and the correspondence between these three is a sought after topic. An increasing amount of research is being conducted on the topics of risk assessment and safety-security interactions [12][16]. However, these topics are still in the earlier stages of developing themselves and thus are often individual studies putting their own emphasis on different possibilities that these types of research bring. With this research we would like to look towards providing a quantitative method to analyse safety-security co-dependency that can form a base that other research can build on rather than analysing one specific case for the purpose of risk or safety-security assessment.

To accomplish this goal, we make use of a Bayesian network [section 2]. Bayesian networks are a way of modelling and visualising situations with multiple probabilistic variables. Bayesian networks are ideal for analysing multiple contributing factors and predicting the likelihood of an outcome and what causes this outcome could have. These properties make Bayesian networks a great modelling tool for safety-security analysis.

By making use of a Bayesian network we model a case and analyse its probabilistic properties. To be able to do this we use GeNIe [3], an academic modelling software that allows one to model a Bayesian network and then change probabilities of variables to see the impact on other probabilistic variables. This way we can simulate all different scenarios without separately having to model them.

Using this model, we introduce *DependencyScore*, a simple formula based on Birnbaum-importance[5] with the goal of ranking events. The events get ranked based on how much impact they have in the Bayesian network on safety and security. Here, the Bayesian network works well again as it gives us quantitative outcomes to work with. This allows us to give values to the impact on events and quantitatively analyse the ranking.

The goal of only using a simple formula is to let this research allow a vast amount of follow-up possibilities and adjustment to fit specific purposes while also providing an overlook insight over these situations.

For the case, we decided on a case of an autonomous surface vessel (ASV). The subject of this case works well within the scheme of safety-security interactions as it contains a cyber-physical system, one of the areas where safety and security blend together the most. For the scope, it also helped that this case had already been used in similar research [7] so that we had some base of the case to work on instead of having to research the case itself more deeply as well, allowing us to focus on the metric.

The deliberation set out above have lead to the following research question to guide the structure of this research:

RQ: *Is DependencyScore an accurate method of ranking the influence of an event on the safety and/or security?*

38th Twente Student Conference on IT Feb. 3rd, 2023, Enschede, The Netherlands.

This will be answered with the help of following sub-questions:

- ***subRQ1***: *What does the Bayesian network of the case look like?*
- ***subRQ2***: *What are the impact values on* `Safe` *and* `Secure` *and DependencyScore value for each of the events?*

The structure of the paper is as follows: In section 2 we will explain the concepts used in this paper to lay out the information necessary for the case study. Section 3 will give an overview of the related works regarding safety-security interactions, Bayesian networks and ASVs. Section 4 will lay out the case study. Section 5 will use the methodology set out in the proposal and walk through what methods were used to answer the research questions set out in the introduction. Section 6 will show the results of the case study. Section 7 will draw conclusions with the help of the case study and research questions and finally, section 8 looks into what can be done on this topic for future research.

2. BACKGROUND

In this section we will give a little more background on the concepts that we will use in this research.

2.1 Bayesian Networks

Many events can have an impact on the safety and security of a system. To analyse the impact of these events and the interactions that happen between safety and security different methods can be used. We will be using Bayesian networks. Bayesian networks illustrate the logical causal associations between different variables. They are able to apply probabilities to these associations and can model what happens when probabilities change.

The essence of Bayesian networks lies in directed acyclic graphs that encode a conditional probability distribution. The nodes of the graph symbolise variables or possible events outcomes and the edges symbolise the relations between these variables. Every node x has a probability distribution C_x on the set $\{0, 1\}$. This distribution is dependent on its n parent nodes and 2 different outcomes and takes form of a table with 2^n rows and 2 columns where the probabilities of every row summed up yields 1.

For example, $A \rightarrow B$ in a Bayesian network would symbolise that the probabilities of variable A influence the probabilities of variable B. Underneath you see a small example of a Bayesian network modelling the probability of the grass being wet.

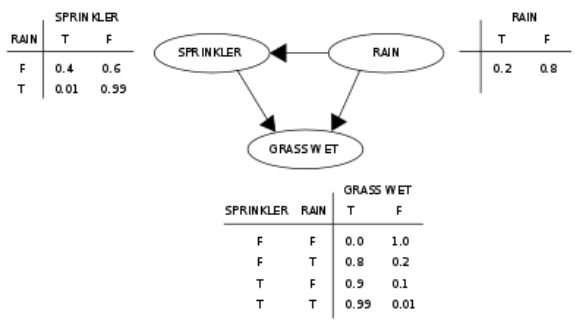


Figure 1. Example Simple Bayesian Network [23]

The reason we use a Bayesian network to analyse our case, besides them working well to visualise any safety-security analysis, is because Bayesian networks allow us to analyse uncertain events with values. Most methods used for analysing safety-security interactions do not have quantitative values to use in a metric.

2.2 Safety-Security Analysis

As we have hinted before, a single event usually has impact on both safety and security. Research classifies what kind of impact and event has on the two with interactions. Today, there are many ways to analyse safety-security interactions. Usually these include analysing the impact of an event on safety and security, resulting in a classification of how it will have impact. These classifications are used to support understanding of the impact.

These are the interactions classified by [14] supported with illustrations from [12].

- **Conditional dependency:** fulfillment of safety events condition safety or vice-versa.

  If event B is conditionally dependent on event A, then event B not happening is only possible if event A has not happened yet: if A occurs then B also occurs.

- **Mutual reinforcement:** safety events contribute to the probability of security events, or vice-versa.

  Event A mutually reinforces event B if B is more likely to happen due to event A and vice-versa.

- **Antagonism:** safety and security events, when considered together, lead to conflicting situations.

  Two events A and B are antagonistic, or conflicting, one cannot occur when the other is occurring and vice-versa. If Event A occurs, Event B automatically doesn't and vice-versa.

- **Independence:** no interaction.

  Two events A and B are statistically independent if $\mathbb{P}(A\&B) = \mathbb{P}(A) \cdot \mathbb{P}(B)$. By assumption, all events are statistically independent. Events that are (mutually) reinforcing can also satisfy this statistical independence requirement.

Nevertheless, usually these analyses of safety-security co-dependence are only done for specific situations, to give insight into probable outcomes and usually it is still a human that decides what course of action this implies.

2.3 The Metric

The metric that we will be using and analysing is a simple one. We will model the Bayesian network in a way that the top events are `Safe` and `Secure`. As all other events have some correlation to these.

In our network each node symbolises an event. To model the consequences of this event to the network and thus the impact of the event to our top events we set evidence. This means we change the probability distribution of this specific node to simulate the event happening or not. In our formula we express this with the numerical values 0 and 1, 1 meaning the event fails and 0 meaning the event is safe. This probability change will propagate through the network and will result in a change in probability distribution for the two top nodes. This means that the top nodes will end with a probability distribution on the

set $\{0,1\}$. We use this probability that the top node is still safe, expressed by $\mathbb{P}(C_y = \{0,1\} \mid C_x = 1)$, in our formula, slightly adjusting Birnbaum importance.

We call the formula *DependencyScore*. In this formula x is the node we are analysing, thus we denote the value for this node as $D(x)$

$$D(x) = I_x(\texttt{Safe}) \cdot I_x(\texttt{Secure})$$

Here $I_x(y)$ is the adjusted Birnbaum-importance of x on y, defined by

$$I_{x^B}(y) = \mathbb{P}(C_y = 0 \mid C_x = 1) - \mathbb{P}(C_y = 0 \mid C_x = 0),$$

with $C_x \in \{0,1\}$ being the stochastic variable that signal the state of node x

As can be seen from the metric, it does not include the different kinds of dependencies we talked about earlier. This is because we also want to see how these dependencies rank interchangeably and if they really matter for the impact of an event.

The metric will yield a value between the 1 and -1. Depending on the value this can insinuate different causes.

- $\boldsymbol{D(x)}$ **is positive:** Both top events (`Safe` or `Secure`) are more likely to be safe or both are more likely to fail based on the evidence set to node x.
- $\boldsymbol{D(x)}$ **is negative:** One of the top events (`Safe` or `Secure`) is more likely to be safe when the analysed event fails.
- $\boldsymbol{D(x)}$ **is zero:** Node x has no influence on either `Safe`, `Secure` or both.
- **Order of magnitude** $\boldsymbol{|D(x)|}$**:** Whether it is positive or negative, the larger $D(x)$, the more impact x has on the top events.

To see if this metric is a good representation of the events impact, we use the metric to rank the events of a case. We will use the Autonomous Surface Vessel (ASV) case from [7] for this.

3. RELATED WORK

In this section we will go over some of the related work in the areas of Safety-Security interactions, Bayesian networks and Autonomous Surface Vessels (ASV).

3.1 Safety-Security interactions

As the world starts to use more and more electronics on a daily basis, the innovation accelerates. Over the past years this has resulted in more focus on Security, especially in the cyber world. But as research realised, safety and security are interconnected, influencing each other. And with the rise of cyber-physical systems, the line between the two often gets blurred [7][19]. Many studies try to make sense of or classify these interactions with different models, Attack and Fault Trees [12][6] or Bayesian networks [17] for example. all applying them to cases or analysing the methods with the goal of being able to make decisions when weighing the two.

3.2 Bayesian networks

Bayesian networks are widely used in research for a range of analysis. They are used from safety and security analysis [8][18][15] to forecasting [21] to population synthesis [20]. Anything with data and probabilities can be modelled into a Bayesian network to predict or get more insight on different scenarios.

3.2.1 *Bayesian metrics*

Several metrics to calculate the impact of a node on the top event in a Bayesian network , not related to safety-security interactions, can already be found in research [22]. For all metrics node (N) and the top event (TE) are used [8].

Birnbaum importance measure[5]: The Birnbaum importance of a node (N) is obtained by computing the difference in probability of top event (TE) for node probability set to 1 and 0 respectively.

$$I_{N^B}(TE) = \mathbb{P}(TE = 1 \mid N = 1) - \mathbb{P}(TE = 0 \mid N = 1)$$

Risk reduction measure: Risk reduction measure demonstrates the effect of the non-occurrence of a basic event on the top event, thereby showing its importance.

$$I_{N^{RR}} = \mathbb{P}(TE = 1) - \mathbb{P}(TE = 0 \mid N = 1)$$

Ratio of variance (RoV) measure: Introduced by [25], this is a measure which uses both prior and posterior failure probabilities of the event to identify the most critical node.

$$RoV_N = \frac{\mathbb{P}^1(N) - \mathbb{P}(N)}{\mathbb{P}(N)}$$

Where $P^1(N)$ represents the posterior failure probability of node N and $P(N)$ represents the prior failure probability of node N.

Our metric is similar to the ones proposed above but using the context of safety-security interactions. We will use the Birnbaum metric on each top event to calculate the impact on one top node but as we are working with two top events, our metric adds an interaction between the two top nodes to analyse the interaction between the two.

3.3 Autonomous Surface Vessels

Since autonomous vehicles have gained popularity, Autonomous surface vessels have been a researched topic. The advantage of autonomous open sea vehicles in comparison to autonomous road vehicles is the lesser amount of scenarios possible due to the nature of the surroundings.

Most ASVs see two modules in the system, the navigation and the collision avoidance. And especially the collision avoidance is researched thoroughly as there are still many scenarios to think of and many methods of analysing these scenarios. Examples of multiple ways of doing collision detection with ASVs that have been researched are Model Predictive Control for Collision Avoidance (MPC COLAV) scheme [9] [11], Fuzzy logic [10] and a Probabilistic Data Association Filter (PDAF) [24], all of these implementations are designed so that they are compliant with COLREGS [1], the international regulations to prevent collisions at sea.

4. CASE STUDY

As far as safety-risks go, industrial practices such as DNVGL-RP0496 [2] already include the possibility of cyber-attacks having influence on physical qualities. For example, if a vehicle suffers a cyber-attack and the control is taken over, the vehicle could suffer physical damages. This is a good example of a potential security breach impacting the safety.

For this specific case we look into the Autosea project (NTNU n.d.), where different disciplines work together to find solutions for Autonomous Surface Vessels (ASV) in maritime applications. One of these solutions is the Telemetron ASV [9]. Equipped with several sensors and

a navigation system it has the ability to autonomously operate at sea [24]. It uses the navigation system to follow a planned path to it's destination and a module of sensors make sure that obstacles on the way are accounted for. If the system senses an obstacle in its planned path the Collision Avoidance System (CAS) is activated. The CAS enables the vehicle to autonomously make manoeuvres to avoid obstacles on its path and come back to its predefined route [11].

5. METHODOLOGY

To recap: We will use the formula *DependencyScore*, a product of two Birnhaum-importances [section 3] which we have further explained in section 2:

$$D(x) = I_x(\texttt{Safe}) \cdot I_x(\texttt{Secure})$$

This section will detail the steps we have taken to answer each of the research questions. First, we will construct a Bayesian network of our case, once we are satisfied with our network and have assigned values to all dependencies we rank all events in terms of impact on safety and security using *DependencyScore* and conclude whether or not this is a good method to rank the impact of events on safety and security.

5.1 subRQ1

From the proposal: ***What does the Bayesian network of the case look like?*** *We will look into the different events possible in the case and how these events are related, resulting in a Bayesian network with* `Safe` *and* `Secure` *as the two top events.This Bayesian network will be constructed in GeNIe.*

Using the research of [7], we constructed a Bayesian network in GeNIe and after a couple of different arrangements of ways to connect the network decided on the network shown in appendix A. Two big design choices we made for this network to noteaere the following:

- *To not split threats apart from their matching prevention fails:* We tried combining respectively all prevention systems and all threats but realised this would result in a loss of information even though it would make the split between safety and security clearer in the network. We decided that the information was more important than the separation.

- *To make two separate sub trees for the difference between wrong messages due to tampering or due to system failure:* The end result would be the same, a wrong message, but we felt that this did not describe the potential hazard behind the cause well enough.

Next, we assigned probabilities to all connections based on estimates. As the goal of this research is not to get the most accurate probability for this case but rather testing the metric, these probabilities may differ when researched further.

5.2 subRQ2

From the proposal: ***What are the impact values on*** `Safe` ***and*** `Secure` ***and DependencyScore value for each of the events?*** *Once we have our Bayesian network in GeNIe, we will run simulations on every different event to see the probabilistic impact on the two top events, applying the formula to each of these runs and making an overview of all these values.*

For the sake of analysing, we decided to use SMILE wrappers [4]. This is an extension for multiple coding languages that allows you to manipulate the network through code, making repetitive tasks more efficient and giving more possibility to analyse the network over different states.

To allow this repetition in our network, all node states are called 'safe' and 'fail'. With SMILE wrappers we determine all values of `Safe` and `Secure` and the *DependencyScore*. This was done with the python code shown in figure 2 compatible with python 3.10.

```
import pysmile
# pysmile_license is your license key
import pysmile_license
def main():
    net = pysmile.Network()
    net.read_file("ASVCase5.xdsl")
    nodes = net.get_all_node_ids()
    for h in range(0, len(nodes)):
        rank = []
        print(nodes[h])
        net.set_evidence(nodes[h], "Safe")
        net.update_beliefs()
        beliefs = net.get_node_value("Safety")
        rank.append(beliefs[0])
        beliefs = net.get_node_value("Security")
        rank.append(beliefs[0])
        net.clear_evidence(nodes[h])
        net.set_evidence(nodes[h], "Fail")
        net.update_beliefs()
        beliefs = net.get_node_value("Safety")
        rank.append(beliefs[0])
        beliefs = net.get_node_value("Security")
        rank.append(beliefs[0])
        net.clear_evidence(nodes[h])
        formula = (rank[2] - rank[0]) * (rank[3] - rank[1])
        print(formula)
        h = net.get_next_node(h)

main()
```

Figure 2. Python code

For overview we left out the values of the top events but these could be printed by adding some extra lines to our code after each updating of beliefs.

The code used outputs the ID of the node and the value of *DependencyScore*.

5.3 RQ

From the proposal: ***Is DependencyScore a accurate me- thod of ranking the influence of an event on the safety and security?*** *Based on the values that our formula gave, we will rank the events in terms of impact on safety and security and analyse whether or not this formula results in an accurate ranking on it's own.*

After running the code, all values are copied to an excel and sorted on *DependencyScore*. This overview was used to compare to the network we made and used to make an analysis of the use of the formula *DependencyScore*. The analysis of the values[B], the network[A] and other results can be found in sections 6 and 7.

6. RESULTS

The final Bayesian network is as can be seen in appendix A. The relations are derived from Fault trees and a UFOI-E, which shows information and energy flows in the system for the sake of safety-security analysis. In our network we assume that both the steering and the acceleration system work similar if not is the same. Thus if someone can penetrate one it means they also have the skills to penetrate the other.

The network has top nodes `Safe` and `Secure`. For this analysis we see these as the ultimate events to happen in both regards, so the `Secure` top event is the scenario of the vessel getting stolen and the `Safe` top event is the scenario of the vessel getting damaged. As shown by the colours of the nodes in the network, some nodes relate more to `Safe` and other to `Secure`. While constructing the network in this way it became apparent that safety and security are never really separate due to the influence they have on each other. From this we can conclude that you cannot analyse one properly without accounting for the other.

Probabilities were assigned on the basis of shallow research. most leaf-events have a probability of failing of about 5%.

On this network, the python code was ran to attain all values. For the sake of overview, the values were left out and only separately reviewed when it was unclear where a result came from. All the *DependencyScore* results are visible in appendix B.

An overview of all values of *DependencyScore* is given in figure 3, where the numbers of the events correspond to the numbers in appendix B.

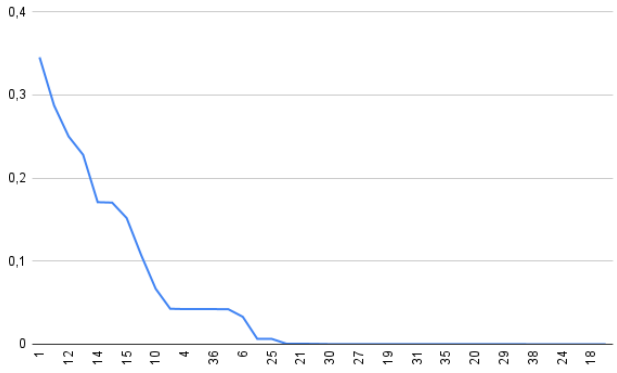


Figure 3. A graph overview of *DependencyScore*

To visualise these values further we also plotted them in a graph with a logarithmic scale to better see the difference in values in figure 4. we used $|D(x)|$ for these graphs. This only mattered for one value of *DependencyScore* as all other values were positive numbers.

This negative number is caused by one of the top events (`Safe` or `Secure`) being more likely to be safe when the analysed event is turned on (or in this case, fails). In this case the probability of the ASV getting stolen decreases when there are extreme weather conditions causing the autopilot to be less reliable.

Looking at the potential scenario where the two top events are not connected, when looking at the numbers of *Dependencyscore*, some of them yield 0. This is easily explained by the fact that these events only had an impact on one of the two top events and *DependencyScore* would thus

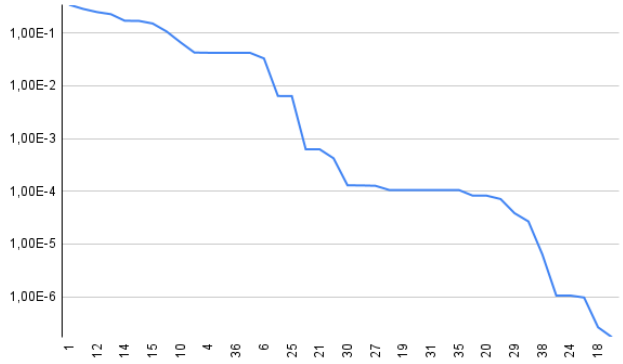


Figure 4. A graph overview of *DependencyScore* in a logarithmic scale

multiply by 0, which yields 0. This also means they would have no impact on the co-dependence of safety and security.

We decided to connect the two top events even though this means that officially there is only one top event. We still consider both of these nodes a top event as they are still both a final outcome.

Before, we concluded that the outcome of *DependencyScore* could differentiate between 1 and -1. Looking at the actual numbers we see that most of them fare close to 0 in the positive direction. The largest impact is about 0.35. As can be expected, generally speaking, events that are closer to the top events have more impact. However, we can also see in the values that the malicious part of the tree in the middle of the network has more impact that coincidental failures.

The node with the highest impact is the 'System compromised' node. As this node is the only node that has direct impact on both top events, this is not surprising. Next to having impact on both events it is the only non-top event that has direct impact on the `Secure` top node. The combination of these two properties makes it unsurprising that this node ranks highest and this likely reflects the real-world scenario.

The bottom of the rank seems to consist of the events that require both a detection failure and a malicious attack, for example, 'GNSS spoofing' and 'GNSS spoofing detection fails', together with accidental system failures. From the essentially deterministic events this was expected as only one event triggering does not have much impact. But we initially expected the accidental failures to have more impact as these have more effect by themselves when triggered.

A possible explanation for this is that these nodes have very little effect on the top `Secure` node and thus, even if they have a lot of effect on the `Safe` node, will not have much impact on our overall analysis.

Joining some of the insights talked about above we take away some main points about our outcomes:

- The closer to the top events, the more impact an event has.
- The events that have impact on both safety and security always rank higher than the ones that only have significant impact on one.
- The better connected a node, the more impact it has.

For the last point we can for example see that the 'Error command to actuator' has more impact than the 'System failure' node. Even though the second node is closer to the top events. This is because the first one is closer related to the 'Malicious message' node which also influences the security side of our network and thus is the 'Error command to actuator' node more connected to the network as a whole.

7. CONCLUSIONS

After analysing the results as shown in section 6, a couple of conclusions can be drawn and some considerations are presented.

Though depth in the network clearly plays a role in the impact it does not play as big of a role as we expected. Well-connected nodes had more impact than purely the depth.

The different kinds of dependencies are generally speaking not visible with this method of analysing. The only thing we can conclude in this area is that usually the impact of an event moves the safety and security in the same direction. As most of the values of *DependencyScore* are positive this means that in most cases $I_x(\texttt{Safe})$ and $I_x(\texttt{Secure})$ are either both negative or both positive, resulting in a positive *DependencyScore* value.

Taking everything into account, this metric works well to give an overview over all nodes and how much they impact both top nodes. However, it does miss some nuances that come with more classifications or more variables. Depth and the kind of dependency can give you a clearer view of the situation and this metric should not be used without looking at these variables. In combination with these, the metric can give a quantifiable way of ranking the events where the network only gives you a qualitative overview. The metric is not all-inclusive but from what we have seen in this case it can provide a solid base.

We have seen some promising results. However, this metric would still need further research to determine the robustness of the metric. From just this case we cannot say if the metric works in more scenarios than just this case. Some examples for different kinds of research following from this can be found in section 8.

8. FUTURE WORK

Some potential following research using the metric proposed in this research are laid out below.

- *Test robustness of the metric:* As also determined in the conclusions above, though the metric works for this case, this doesn't prove that the metric will work in other situations.

To be able to use this metric on a larger scale, research needs to be done to define the robustness of the metric. Possibly this could be done using data analysis on this specific case, changing the probabilities to see if the metric holds.

- *Test metric on a specific dependency:* In our case, almost every value of $D(x)$ was positive. This is likely because of the kinds of dependencies that show up in our case. It would be interesting to see what would happen if you would test this metric on a case study with a lot of antagonisms for example, where we would expect more negative $D(x)$ values.

- *Adding more variables to the metric:* This metric is not nuanced enough to be used solely on it's own, in the conclusions we deduced that analysing other variables was necessary to get a good overview. One way to combat this would be to identify the variables that need to be taken into account, like depth and dependency, and integrating them into the formula to still have one formula that can be used to analyse the impact of events in safety-security co-analysis.

- *Formulating a framework that includes the metric as well as a way to analyse the other variables:* Another way to combat the issue of this formula not being able to describe enough on it's own, is not to add variables to the formula, but to create a bigger framework that includes the metric as well as other analysing steps to close the gap.

Next to these we also propose some similar research topics in the field not using the metric here.

- *Mathematical definitions dependencies:* Right now, with the exception of independency, there are no mathematical definitions for the dependencies. This makes working with these dependencies a somewhat subjective and never-ending matter. Giving the dependencies mathematical definitions can help to unify research being done on the topic as well as build a solid basis for more complex cases.

- *Modelling dependencies into a formalism:* As of now, there are no formalisms that allow us to read the different kinds of dependencies directly from the model, this is one of the reasons that complex cases are hard to dissect. Introducing a modelling technique that allows us to read the dependencies directly out of the model would help with these complex cases.

Of course this proposed future research could later be combined with research about the metric again to give an even better basis for analysing safety-security interactions.

References

[1] Colregs - international regulations for preventing collisions at sea, 1972. URL http://www.mar.ist.utl.pt/mventura/Projecto-Navios-I/IMO-Conventions%20(copies)/COLREG-1972.pdf.

[2] Dngvl-rp-0496 cyber security resilience management for ships and mobile offshore units in operation, 2016. URL https://www.gard.no/Content/21865536/DNVGL-RP-0496.pdf&sa=U.

[3] Bayesfusion. Genie modeler, 2015. URL https://www.bayesfusion.com/genie/.

[4] Bayesfusion. Smile engine, 2016. URL https://www.bayesfusion.com/smile/.

[5] Z. Birnbaum. On the importance of different components in a multicomponent system. 2:24, 05 1968.

[6] C. E. Budde, C. Kolb, and M. Stoelinga. Attack trees vs. fault trees: Two sides of the same coin from different currencies. In A. Abate and A. Marin, editors, *Quantitative Evaluation of Systems*, pages 457–467, Cham, 2021. Springer International Publishing. ISBN 978-3-030-85172-9.

[7] N. H. Carreras Guzman, D. Kufoalor, I. Kozine, and M. Lundteigen. Combined safety and security risk analysis using the ufoi-e method: A case study of an autonomous surface vessel. 09 2019. doi: 10.3850/978-981-11-2724-3_0208-cd.

[8] P. G. George and V. Renjith. Evolution of safety and security risk assessment methodologies towards the use of bayesian networks in process industries. *Process Safety and Environmental Protection*, 149:758–775, 2021. ISSN 0957-5820. doi: https://doi.org/10.1016/j.psep.2021.03.031. URL https://www.sciencedirect.com/science/article/pii/S0957582021001452.

[9] I. Hagen, D. Kufoalor, E. Brekke, and T. Johansen. Mpc-based collision avoidance strategy for existing marine vessel guidance systems. pages 7618–7623, 05 2018. doi: 10.1109/ICRA.2018.8463182.

[10] Y. Hu, X. Meng, Q. Zhang, and G.-K. Park. A real-time collision avoidance system for autonomous surface vessel using fuzzy logic. *IEEE Access*, 8:108835–108846, 2020. doi: 10.1109/ACCESS.2020.3001626.

[11] T. A. Johansen, T. Perez, and A. Cristofaro. Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3407–3422, 2016. doi: 10.1109/TITS.2016.2551780.

[12] C. Kolb, S. M. Nicoletti, M. Peppelman, and M. Stoelinga. Model-based safety and security co-analysis: a survey. *CoRR*, abs/2106.06272, 2021. URL https://arxiv.org/abs/2106.06272.

[13] S. Kriaa, M. Bouissou, F. Colin, Y. Halgand, and L. Pietre-Cambacedes. Safety and security interactions modeling using the bdmp formalism: Case study of a pipeline. In A. Bondavalli and F. Di Giandomenico, editors, *Computer Safety, Reliability, and Security*, pages 326–341, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10506-2.

[14] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand. A survey of approaches combining safety and security for industrial control systems. *Reliability Engineering System Safety*, 139:156–178, 2015. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2015.02.008. URL https://www.sciencedirect.com/science/article/pii/S0951832015000538.

[15] D. Lichte and K.-D. Wolf. Bayesian network based analysis of cyber security impact on safety. pages 1502–1509, 01 2019. doi: 10.3850/978-981-11-2724-3_0288-cd.

[16] A. Mashkoor, A. Egyed, and R. Wille. Model-driven engineering of safety and security systems: A systematic mapping study, 2020. URL https://arxiv.org/abs/2004.08471.

[17] S. Pirbhulal, V. Gkioulos, and S. Katsikas. Towards integration of security and safety measures for critical infrastructures based on bayesian networks and graph theory: A systematic literature review. *Signals*, 2(4):771–802, 2021. doi: 10.3390/signals2040045. URL https://www.mdpi.com/2624-6120/2/4/45.

[18] J. Shin, H. Son, R. Khalil ur, and G. Heo. Development of a cyber security risk model using bayesian networks. *Reliability Engineering System Safety*, 134:208–217, 2015. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2014.10.006. URL https://www.sciencedirect.com/science/article/pii/S0951832014002464.

[19] M. Steiner and P. Liggesmeyer. Combination of safety and security analysis - finding security problems that threaten the safety of a system. 2016. URL http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-43604.

[20] L. Sun and A. Erath. A bayesian network approach for population synthesis. *Transportation Research Part C: Emerging Technologies*, 61:49–62, 2015. ISSN 0968-090X. doi: https://doi.org/10.1016/j.trc.2015.10.010. URL https://www.sciencedirect.com/science/article/pii/S0968090X15003599.

[21] S. Sun, C. Zhang, and G. Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):124–132, 2006. doi: 10.1109/TITS.2006.869623.

[22] M. van der Borst and H. Schoonakker. An overview of psa importance measures. *Reliability Engineering System Safety*, 72(3):241–245, 2001. ISSN 0951-8320. doi: https://doi.org/10.1016/S0951-8320(01)00007-2. URL https://www.sciencedirect.com/science/article/pii/S0951832001000072.

[23] Wikipedia. Bayesian network, Jun 2022. URL https://en.wikipedia.org/wiki/Bayesian_network.

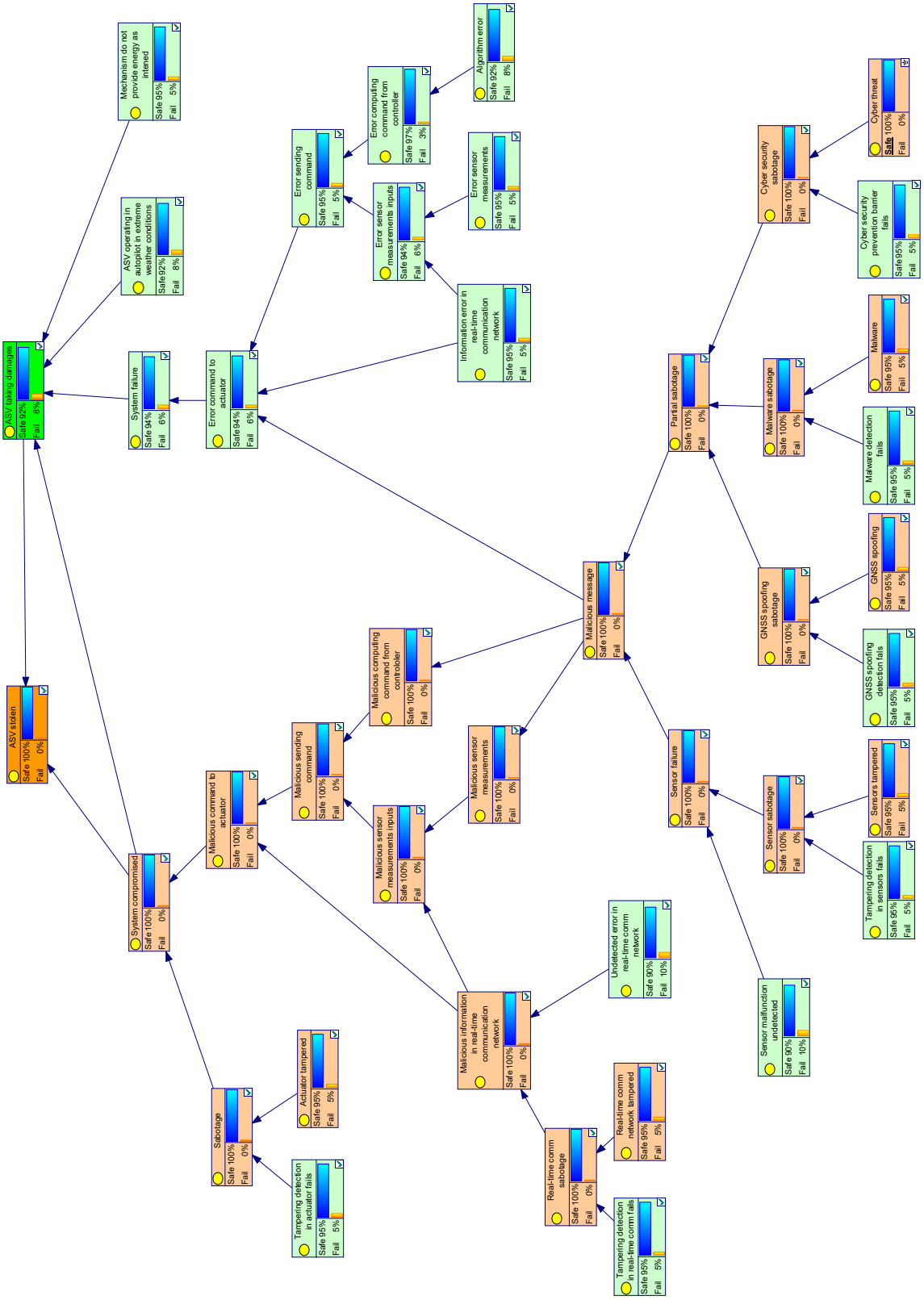
[24] E. Wilthil, A. Flåten, and E. Brekke. *A Target Tracking System for ASV Collision Avoidance Based on the PDAF*, pages 269–288. 05 2017. ISBN 978-3-319-55371-9. doi: 10.1007/978-3-319-55372-6_13.

[25] E. Zarei, A. Azadeh, N. Khakzad, M. M. Aliabadi, and I. Mohammadfam. Dynamic safety assessment of natural gas stations using bayesian network. *Journal of Hazardous Materials*, 321:830–840, 2017. ISSN 0304-3894. doi: https://doi.org/10.1016/j.jhazmat.2016.09.074. URL https://www.sciencedirect.com/science/article/pii/S0304389416308950.

APPENDIX

A. BAYESIAN NETWORK ASV CASE

You can click here to download the network as a pdf.



B. NUMERICAL VALUES RANK

	Name	D(x)
1	System_compromised	0,3451377532
2	Malicious_command_to_actuator	0,2871068071
12	Sabotage	0,2499214004
11	Malicious_sending_command	0,2277994328
14	Malicious_sensor_measurements_inputs	0,1708467543
3	Malicious_information_in_real_time_communication_network	0,1703288144
15	Malicious_computing_command_from_controller	0,1517580129
16	Malicious_sensor_measurements	0,107195239
10	Malicious_message	0,06668800985
8	Partial_sabotage	0,04266146699
4	Cyber_security_sabotage	0,04223611199
34	GNSS_spoofing_sabotage	0,04223611199
36	Malware_sabotage	0,04223611199
17	Sensor_failure	0,04212297654
6	Real_time_comm_sabotage	0,03297550857
26	Error_command_to_actuator	0,006407556814
25	System_failure	0,006386962774
13	Actuator_tampered	6,25E-04
21	Tampering_detection_in_actuator_fails	6,25E-04
22	Sensor_sabotage	4,21E-04
30	ASV_operating_in_autopilot_in_extreme_weather_conditions	1,30E-04
37	Information_error_in_real_time_communication_network	1,29E-04
27	Error_sending_command	1,27E-04
5	Cyber_security_prevention_barrier_fails	1,06E-04
19	Cyber_threat	1,06E-04
32	Malware	1,06E-04
31	GNSS_spoofing	1,06E-04
33	GNSS_spoofing_detection_fals	1,06E-04
35	Malware_detection_fails	1,06E-04
7	Tampering_detection_in_real_time_comm_fails	8,24E-05
20	Real_time_comm_network_tampered	8,24E-05
39	Error_sensor_measurements_inputs	7,12E-05
29	Mechanism_do_not_provide_energy_as_intened	3,84E-05
28	Error_computing_command_from_controller	2,67E-05
38	Error_sensor_measurements	6,34E-06
23	Sensors_tampered	1,05E-06
24	Tampering_detection_in_sensors_fails	1,05E-06
40	Algorithm_error	9,65E-07
18	Sensor_malfunction_undetected	2,63E-07
9	Undetected_error_in_real_time_comm_network	1,70E-07