# Simulating ecosystem-level cybersecurity for the future generation of critical infrastructures

Exploring the impact of cyber-defensive strategies on critical infrastructures through agent-based modelling

Xander de Ronde

TU Delft

*This page is intentionally left blank*

# Simulating ecosystem-level cybersecurity for the future generation of critical infrastructures

Master thesis submitted to Delft University of Technology

in partial fulfilment of the requirements for the degree of

**MASTER OF SCIENCE**

in Complex Systems Engineering and Management

Faculty of Technology, Policy and Management

by

Xander Elco Jan de Ronde

Student number: 4170334

To be defended in public on September 7th, 2018

## Graduation committee

| | |
|---|---|
| Chairperson | : Prof.dr.ir. P.H.A.J.M. van Gelder, Section Safety and Security Science |
| First Supervisor | : Dr.ir. W. Pieters, Section Safety and Security Science |
| Second Supervisor | : Dr. M.E. Warnier, Section Systems Engineering & Simulation |

*This page is intentionally left blank*

# Preface

This report encompasses the final product of many months of hard work on the master's thesis in fulfilment of the Master of Science degree in Complex Systems Engineering and Management at Delft University of Technology. This master's thesis brings together research skills that I was able to develop throughout my academic career. Throughout this project I found myself deeply engaged in conducting research and analysis on a scientific and societal problem that, over the course of this project, became increasingly prevalent. Throughout both my BSc and MSc programmes I found myself endeared by the task of designing security solutions for critical infrastructures. Recent technological and societal developments have steadily emphasised my interest in the subject matter and pushed me towards further improving my work. Additionally, simulation modelling and specifically agent-based modelling were topics that I thoroughly enjoyed over the course of both my BSc and MSc programmes. I am thankful that I have been able to find a project that allowed me to research a topic I am deeply interested in with a method that I also thoroughly enjoy. Furthermore, I am proud of the work I have been able to put in and the results that followed.

This thesis is written in an attempt to enhance the academic state of the art by establishing and using a framework for cybersecurity analysis of critical infrastructures. As such, this report is written for an audience of policymakers, engineers and academics who are interested in designing cybersecurity solutions for critical infrastructures. However, given the multidisciplinary research discipline that overarched this study, the report can also prove interesting for other readers. This is especially true given the recently increased societal relevance and awareness of the problem this study seeks to address. I hope that my work and passion contributed to some extent to the state of the art.

*This page is intentionally left blank*

# Acknowledgements

Handing in this thesis marks the end of my academic career at Delft University of Technology. This university and specifically the faculty of Technology, Policy and Management have always inspired me to go the extra mile. While this project was an individual assignment, I would like to extend my gratitude for those who have supported and assisted me throughout the long and endearing process of writing this dissertation and my academic career in general.

Firstly, I would like to thank the graduation committee, Wolter Pieters, Martijn Warnier and Pieter van Gelder, for the feedback they have provided over the course of this project. Their critical reflections on not only the scientific narrative, but also minor descriptions and inaccuracies, helped me greatly improve the quality of my work. The committee helped me stay on track when my ambition and motivation would pursue me to extend the scope and scale of my research activities beyond what was manageable.

Secondly, I would like to thank fellow students I regularly convened with to work on our theses. These study sessions allowed us to discuss possible solutions to problems in conducting a large research project, dealing with a modelling challenge and so on. By helping others I found myself commonly finding possible improvements for my own work as well after critically reflecting on my own progress.

Third and last, but certainly not least, I would like to express my gratitude towards my great family and friends who have tolerated, supported and motivated me while I spent many hours, days and weeks on tackling challenges in my academic work. Without your kindness, support and trust over the course of my time at this university these achievements would have been out of reach.

*Xander de Ronde*

*Delft, August 2018*

*This page is intentionally left blank*

# Executive summary

**The main problem** that this study seeks to address is the increased threat posed to modern-day society by cyberattacks on critical infrastructures. The transition from monolithic, closed critical infrastructure systems towards decentralised and open networks has enabled a significant increase in efficiency, at the cost of increased susceptibility to cyberattacks. Securing critical infrastructures against threats that exploit cyber-infrastructure to inflict large-scale physical damage or even loss of life has become a significant objective for both scientific research and public policy analysis.

**The aim of this study** is to explore the effects of coherent defensive strategies on cyber-defensive behaviour in an ecosystem of critical infrastructures. The academic state of the art prescribes a shared desire to explore the effects of coherent defensive strategies that help secure the overall ecosystem of critical infrastructures more effectively. By integrating generalisable elements applicable to most critical infrastructure systems, ecosystem-level behaviour can be identified. This led to the following main research question:

> *How do cyber-architectural elements and defensive strategies influence exposure to cyber-threats within the cybersecurity ecosystem of critical infrastructures, and how can infrastructure operators effectively mitigate consequences from cyber-incidents?*

**The research objectives** involve establishing a framework for analysing a cybersecurity ecosystem for critical infrastructures. To achieve this, three main objectives are formulated: (i) specification of concepts that form an ecosystem model for critical infrastructure systems by identifying generalisable elements, (ii) formalisation and implementation of an agent-based model capable of exploring the effects of different configurations for integrated defensive strategies and (iii) deriving experimentation results into observed emergent patterns and best practices for defensive strategies.

**The main method** used to support these objectives is agent-based modelling, which simulates systems as collections of distributed entities capable of individual decision-making based on a set of states, rules and actions. Agent-based models are capable of simulating complex systems through means of simple actions on the level of individual agents.

**The key interaction** this study seeks to address takes place between critical infrastructure operators, cyberattackers and users who are critical to infrastructure operation. Critical infrastructure operators are tasked with securing the functional operation of critical infrastructures, for which involves thwarting cyberattacks while not hindering users. Cyberattackers seek to inflict damage on critical infrastructure nodes by launching cyberattacks. Users only interact with the ecosystem in an auxiliary manner, as they provide user traffic that can hinder infrastructure operation.

**The core elements** identified that form the ecosystem are cyber-architectural concepts, cybersecurity concepts and behavioural elements. The main cyber-architectural concepts are *dependencies* that affect infrastructure node operation beyond the targets of cyberattacks alone and the severity of *consequences* that emphasise the desire for effective defensive strategies. The main cybersecurity elements involve the *types of attackers* that impact the type and focus of attacks conducted, the *types of cyberattacks* and *control mechanisms* that constitute defensive strategies. The main behavioural elements relate to *situational awareness* that limits interaction due to limited rationality, specifically the degree of *operability* and associated degree of *perceived operability* that determine which defensive decision is made.

**The implemented simulation model** incorporates the core elements and operationalised interaction that takes place within the ecosystem. This simulation model can be used to assess system behaviour under different configurations for scenario parameters or defensive strategy designs. These

parameters influence the frequency and impact of certain events, such as a cyberattacks, and simulate the chance for these events to be successful. This allows for model use to extend beyond static analysis and facilitates exploration of deep uncertainty surrounding the ecosystem.

**The experimental design** is based on exploration of system behaviour under deep uncertainty by conducting Exploratory Modelling and Analysis (EMA). The exploratory approach fits the chosen research design, as ecosystem-level analysis is built on tentative parameter values. Instead, a wide range of scenarios was created to suppress sensitivity towards specific parameter values, with variations applied across parameters such as the influence coefficient associated with dependencies. Experiments are defined as the combination of a scenario and a defensive strategy design. Each defensive strategy design was analysed using the same set of scenarios, resulting in a complete set of experiments that could be used to explore the robustness of each defensive strategy design.

**The main findings** indicate that cyber-resilience of critical infrastructures is tied to three main emergent tendencies:

1. When there is insufficient awareness of threats, the impact of those attacks is amplified by the delay in deployment of responsive mechanisms, as attacks are not dealt with in a timely manner. This occurred most prominently for signature-based intrusion prevention and detection mechanisms.
2. When there is an unmanageable rate of false alarms, responsive mechanisms can end up inflicting more damage than utilising no responsive mechanisms at all. Defensive strategies that overstate the presence of attacks can therefore have a negative impact. This tendency was particularly common for anomaly-based intrusion prevention and detection mechanisms.
3. Preventing cyberattacks by design delivers robust performance, whereas experiments tailored towards preventing false alarms bear the most immediate benefits, since attack events are uncommon and unnecessary responses can be initiated at any point. This tendency involves the way defensive decisions carry over to other infrastructure nodes through dependencies.

**The main limitations and assumptions** are the direct consequences from designing an ecosystem-level model. Because all elements are forced to be generalisable for most critical infrastructure systems, the specification of several concepts remains rather abstract. Assessing different defensive strategies from an ecosystem perspective does not produce results directly related to implementable policy, as that requires more thorough specification. The main assumption that influences model performance directly followed this limitation, as attacks in the ecosystem are always surmountable. Insurmountable consequences, as observed in very rare occasions in the real-world, are a deterrent that impacts the extent of losses incurred. This metric is kept relatively abstract in this study, which delimits the direct interpretability of model outcomes.

**The main implications and contributions** put forward by this study are nested in the framework created to support the simulation model. Given the limitations and assumptions, the most important contribution is the conceptual framework of cybersecurity for an ecosystem of critical infrastructures. Since data analysis within the light of this approach does not contribute any new knowledge, the more important facet of this study is establishing future pathways for research and policy engineering. As such, this study contributes to the academic state of the art by proving that simulation modelling of critical infrastructures in a coherent ecosystem is possible and can help establish which behavioural patterns are expected under different policy configurations. By establishing a framework that incorporates all generalisable and applicable concepts, future research and analysis can expand on the foundation of this study.

# Table of contents

# List of figures

# List of tables

# Definitions

*Table 0-1: List of definitions*

| *Term* | *Description* |
|---|---|
| *Critical infrastructure (CI)* | An infrastructure for which the unhindered operation of critical infrastructures is vital for the functioning of crucial elements of society. Another term used is Networked Control System (NCS), which is a more abstract concept of interconnect subsystems to which critical infrastructures belong. |
| *Infrastructure node* | A distributed or decentralised component of a larger critical infrastructure system. This can entail any control system, sensor, communication |
| *SCADA systems* | Supervisory Control And Data Acquisition systems are central entities that assert control over parts of critical infrastructure networks. They are part of older, legacy infrastructures which relied on centralised architecture to ensure secure operation. |
| *Infrastructure operability/operation* | The degree by which an infrastructure is capable of functioning when compared to normal operation. Internal inoperability can be caused by attacks, erroneously blocked user traffic or responsive mechanisms. External inoperability can only be caused by dependencies. Infrastructure inoperability directly contributes to losses incurred. |
| *Dependency* | A functional directional connection between two infrastructure nodes that directly inflicts a degree of inoperability to a dependent node based on disruptions in the origin node. |
| *Control mechanism* | A countermeasure in place to either *prevent*, *detect* or *respond to* intrusions. |
| *Defensive strategy* | A configuration for a set of control mechanisms. |
| *Impact assessment* | The process by which an infrastructure operator attempts to detect intrusions. If an alarm is generated, the impact assessment changes by the expected impact of the detected type of attack. |
| *Complex adaptive systems (CAS)* | A paradigm of systems thinking that perceives systems as “a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing.” (Waldrop, 1992). |
| *Agent* | An entity in an agent-based model that is capable of autonomous, independent decision-making based on a set of states, rules and actions. |
| *Infrastructure node operators (defenders)* | Agents who are tasked with maintaining secure infrastructure node operability. |
| *Cyberattackers* | Agents who seek to inflict damage to infrastructure nodes or the environment. |
| *Users* | Agents who make use of infrastructure nodes. |
| *Chaos* | The complex behaviour resulting from variations in initial conditions that can occur through a set of model runs. Since each run uses different random number generator seeds, the structure of a model as well as the interaction that takes place will vary across the set of experiments. |
| *Threat landscape* | The collection of all active threats to the core operation of a critical infrastructure. Infrastructure operators make an assessment of the |

| | |
|---|---|
| | threat landscape by assessing which elements pose a current threat to the system. |
| *Threat attractiveness/attacker utility* | How attractive a possible target node is to an attacker attempting to launch an attack. This is composed by the economic and physical losses associated with node inoperability and attacker preference for economic and physical damage. |
| *Situational awareness* | The degree to which an entity of the system is aware of the true state of a given element. An attacker wants to maximise their utility based on their degree of situational awareness and node operators seek to make correct defensive decisions against threats they are aware of. |
| *Repetition/iteration* | A single full simulation of the entire desired time frame for the model under a single set of parameter values. Multiple repetitions for one experiment are multiple model simulations with a single set of parameter values, used to suppress the impact of chaos. |
| *Tick* | A single time step in an agent-based model by which model procedures are conducted and output parameters are tracked. |
| *Evaludation* | A framework put forth by Augusiak, Van den Brink, and Grimm (2014) that can help ensure validity of outcomes for exploratory simulation models operating under deep uncertainty. Instead of focusing purely on statistical outcomes and their observed real-world counterparts, this approach focuses on the thoroughness of concepts and behaviour associated with the model. |

*This page is intentionally left blank*

# 1 Introduction

Over the years, modern society has grown increasingly reliant on uninterrupted operation of critical infrastructure (CI) systems. Critical infrastructures are involved in many different societal tasks, such as ensuring efficiency on the electricity grid, public safety in national health infrastructure and flood protection systems. Technological developments and emergent smart city thinking have substantially increased the burden on critical infrastructures (Department of Homeland Security, 2015). This originates from the shift from monolithic, single-purpose systems towards large networks of distributed and heterogeneous system components (Ericsson, 2010). While these changes have led to a drastic increase in the capabilities of critical infrastructures, an unwanted and in many cases unexpected consequence is their susceptibility to attacks in both cyber and physical domains (Brezhnev, Kharchenko, Manulik, & Leontiev, 2018; Brown, Carlyle, Salmerón, & Wood, 2006; Farwell & Rohozinski, 2011). Given the critical role of these systems in society, the desire emerges to establish effective ways to mitigate risk. The aim of this thesis is to assess the effectiveness of defensive strategies for critical infrastructure systems. This chapter provides an introduction to the societal and scientific relevance of securing critical infrastructures. The first section details definitions of critical infrastructures and the essence of cybersecurity challenges within this domain. The second section provides insight in the scientific state of the art in critical infrastructure security. The third section derives the main research direction for this study. The fourth section lists the approach taken to answer the main research question. Fifth and last, the overall structure of this thesis is presented.

## 1.1 Critical infrastructures and cybersecurity

While there are multiple definitions of critical infrastructures, the terms ‘vitality’ or ‘criticality’ are almost universally included. The distinction between infrastructures and critical infrastructures lies in these very notions: the unhindered operation of critical infrastructures is vital for the functioning of crucial elements of society (Moteff, Copeland, & Fischer, 2003; Van der Lei, Bekebrede, & Nikolic, 2010). Even minor disruptions in the power grid could result in massive blackouts, potentially crippling a nation’s economy and other crucial sectors depending on the stability of the electricity grid (Farwell & Rohozinski, 2011; Romanosky & Goldman, 2016).

The shift towards decentralised and distributed networks of heterogeneous system components led to cyber-physical systems capable of connecting more means of gathering information in one environment (Amin, Litrico, Sastry, & Bayen, 2013a; Khurana, Hadley, Lu, & Frincke, 2010). However, this single environment is now subject to different standards and protocols, constraining agility of the system. Responsibility to secure these components is often distributed, leading to a loosely coupled set of security requirements that do not necessarily translate into effective countermeasures (Neuman, 2009). In turn, this resulted in an increased reliance on the presence and availability of all connected components (Baiardi, Suin, Telmon, & Pioli, 2006; Sandberg, Amin, & Johansson, 2015). The shifted ecosystem for critical infrastructures led to a stark increase in impact and frequency of cyberattacks on civilian targets, as highlighted by numerous attacks on electricity grids, hospitals or water management systems. Recent examples include coordinated attacks on the Ukrainian power grid in 2015 or the Stuxnet attacks on Iranian nuclear facilities. The former left to hundreds of thousands of civilians without power, severely damaging the regional economy (Lee, Assante, & Conway, 2016; Liang, Weller, Zhao, Luo, & Dong, 2017). The latter, Stuxnet, served as a wake-up call for engineers around the world, setting in motion a scramble for knowledge on how to defend against targeted attacks on critical infrastructure assets (Farwell & Rohozinski, 2011; Karnouskos, 2011).

The discipline of cybersecurity traditionally revolves around information risk in organisations, where actors seek to maximise their security spending based on a degree of acceptable risk. Critical infrastructures do not operate like this, there is simply no form of acceptable risk. Securing critical infrastructures calls for a different approach, where networks need to studied in unison. The immense societal consequences from critical infrastructure failures and their exposure to cyberattacks stress the need for security by design in order to effectively detect and thwart intrusions (Fairley, 2016; Karnouskos, 2011). The societal importance of critical infrastructure systems calls for effective defensive strategies, as existing approaches cannot cope with the increased sophistication of the threats they are facing (Amin, Litrico, Sastry, & Bayen, 2013b; Department of Homeland Security, 2015). Establishing up-to-date, consistent defensive strategies requires understanding interactions within the ecosystem of cyberattacks on critical infrastructures (Fairley, 2016).

## 1.2 Academic state of the art

Before a research direction can be specified, a scoping literature review was conducted. The literature review, of which the summary is found in Appendix A, identifies several academic approaches to cybersecurity for critical infrastructures. The main purpose of the literature review is to establish the state of the art of scientific literature, which helps identify the gap of knowledge that this study can address. The state of the art involves four main concepts that pose primary research directions for securing critical infrastructures against cyberattacks. These concepts are as follows:

The first recurring theme in academic literature is the architectural complexity of cyber-physical critical infrastructures. Increased accessibility and interconnection of system components have facilitated great improvements in productivity of critical infrastructures (Cárdenas et al., 2011). On the other hand, these very changes have formed vulnerabilities: an increase in connectivity increased the attack surface for malicious actors (Hahn, Ashok, Sridhar, & Govindarasu, 2013; Karnouskos, 2011; Sandberg et al., 2015). The architectural complexity requires a specific line of thinking and complicates traditional cybersecurity approaches (Brezhnev et al., 2018; Department of Homeland Security, 2015; Karnouskos, 2011).

The second recurring theme is the heterogeneous nature of both cyberattacks and the control mechanisms that are designed to prevent them. The variety in possible attack vectors has muddied the waters in identifying attainable security goals (Department of Homeland Security, 2015; Fairley, 2016). Conversely, different types of control systems apply specific solutions that work well within their individual domain, but would not translate well into generalisable security policy (Formby, Durbha, & Beyah, 2017). Securing critical infrastructures strikes a balance between effective, specific solutions and coherent, shared defensive strategies.

The third theme identified is the limited degree of rationality and dependency on accurate information. Traditional cybersecurity models typically assess the ecosystem as either exclusively rational or exclusively irrational optimisation problems, whereas the real-world situation revolves around making decisions based on available data (Alcaraz & Lopez, 2013; Amin et al., 2013a; Liu, Stefanov, Hong, & Panciatici, 2012; Teixeira, Amin, Sandberg, Johansson, & Sastry, 2010).

The fourth and last recurring theme is the overall need for coherent defensive strategies. The failure to account for cybersecurity issues for critical infrastructures makes the task a lot more difficult (Clark, Panguluri, Nelson, & Wyman, 2017). Desires for security by design are logical, but too late for irreplaceable existing infrastructure. As a result, there is a need for coherent strategies to effectively implement mechanisms that thwart and mitigate cyberattacks (Cárdenas et al., 2011; Neuman, 2009).

## 1.3 Knowledge gap

While there are some stark and subtle differences in academic literature on securing critical infrastructures, there is a clear need for further research in this field. The realisation that security incidents can result in large-scale consequences has led to researchers scrambling for new methods and insights. Several recurring themes of academic literature were identified in section 1.2, yet are typically described in isolation. The knowledge gap therefore involves a lack of understanding of ecosystem-level system behaviour when all four elements are taken into account. While plentiful literature exists on each of these elements, academics have thus far not been able to explore different scenarios for defensive strategies.

In order to frame these themes within the light of detail these require, a basic conceptual overview of the system was created. This conceptual overview, shown in Figure 1-1, serves as the starting point for this study and will be expanded throughout the report. The only elements included are key entities to infrastructure operation, as well as the general interaction taking place among them.

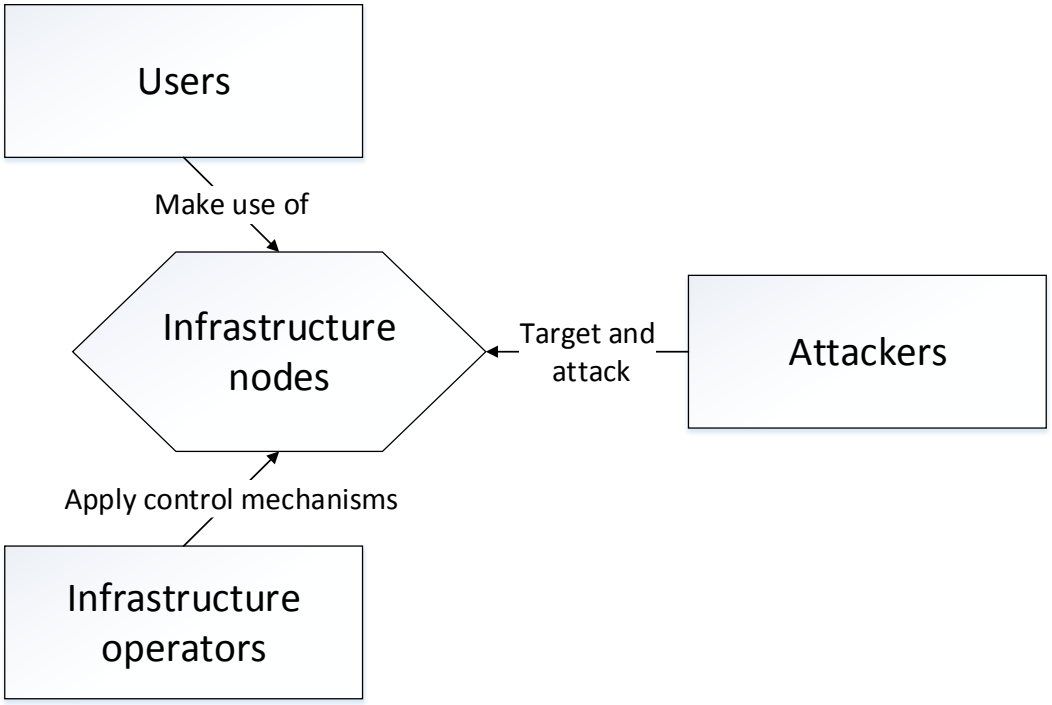


*Figure 1-1: Conceptual overview of key ecosystem entities and interactions*

After all core concepts are addressed and included in the conceptual overview, a fitting basis for exploration of ecosystem-level behaviour will have been established. As shown in the overview, infrastructure nodes play a central role in understanding the cybersecurity ecosystem for critical infrastructures. The main interaction is related to infrastructure operators and to attackers, as these make use of various defensive and offensive strategies to achieve their primary goals. In that sense, users maintain a more auxiliary role to the problem. Ecosystem specification will therefore mainly emphasise concepts related to infrastructure operators and attackers.

Recalling the main knowledge gap, the four themes converge in critical infrastructure cybersecurity, yet little effort is made to model all of these aspects in unison. To this end, a main research question is formulated to specify the direction for this study. The main research question forms the backbone of this study. As established earlier, the aim is to explore and assess system behaviour and performance across a variety of different scenarios for defensive strategies. The main research question for this agent-based modelling study is formulated as follows:

*How do cyber-architectural elements and defensive strategies influence exposure to cyber-threats within the cybersecurity ecosystem of critical infrastructures, and how can infrastructure operators effectively mitigate consequences from cyber-incidents?*

## 1.4 Research approach

As stated before, the main research question will be answered by conducting a comprehensive agent-based modelling study. First, main research objectives will be formulated. These research objectives indicate actionable tasks to be carried out to formulate an answer to the main research question. Secondly, the conceptual scope is detailed, specifying the conditions that serve as boundaries for the research project. Thirdly, the general research approach will be laid out. Fourthly, sub-questions are formulated, which together can form a coherent answer to the main research question.

### 1.4.1 Research objectives

The main aim of the model is to establish the effectiveness of defensive strategies for critical infrastructures. Exposure to cyber-risk within critical infrastructures has led to a growing desire for coherent security policies. By creating a simulation model that incorporates multiple interdependent critical infrastructure nodes, effects of cascading failures can be explored. Failures in individual infrastructure nodes might carry over to directly or indirectly connected nodes. This highlights the requirement for coherent security policies, which are to be explored using an agent-based model. Individual decision-making capabilities with different perceptions of current threats for a larger number of agents within an infrastructure are traditionally not included in cyber-risk models. Agent-based modelling allows for these effects to be included, while maintaining graph-based spread of cyberattacks through networks. The resulting agent-based model is based on the direct and indirect spread of consequences from cyberattacks, paying particular attention to cyber-defensive strategies. Given these descriptive goals, the following objectives are formulated:

Objective 1: *Specify and conceptualise an ecosystem model for CI systems by establishing elements that relate to each core concept.*
Objective 2: *Formalise and specify this ecosystem into a fully-fledged agent-based model capable of simulating different configurations for integrated defensive strategies.*
Objective 3: *Derive the simulation results into emergent patterns and best practices for effective cyber defensive strategies for critical infrastructures.*

### 1.4.2 Scoping and constraints

Conducting an agent-based modelling study can turn out to be a demanding and complex endeavour. The scope of a simulation model can extend far beyond the original purpose of a study. Devising simulation models involves translating reality into representative concepts, while maintaining manageability of the entire research process. The concepts touched upon in section 1.2 are all crucial to understanding critical infrastructures. The level of abstraction applied to the simulation model has to correspond with the formulated research objectives.

The approach for this study involves modelling a representative selection of critical infrastructure elements. Elements to receive particular emphasis directly relate to the estimation of operability in sections of the infrastructure network and how inoperability is carried over to other portions of the model. This involves both attackers' and defenders' understanding of a particular situation. Less interesting for the purposes of this study are representative indications of many different infrastructures in one ecosystem. Instead, the model will include generalisable elements found in each infrastructure, yet remain extensible to incorporate infrastructure-specific elements. The key model inputs of the model to different configurations of defensive strategies in multiple scenarios.

This implies that details surrounding specific interaction between vulnerabilities and mechanisms used for attack and defence are not as interesting. The implementation of such mechanisms will rely on real-world effectiveness of attack vectors and control mechanisms. By making this concession, the simulation model can include more refined and generalisable decision-making models that are more likely to yield representative results in terms of sustained losses.

### 1.4.3 Agent-based modelling

A comprehensive agent-based modelling study will be conducted, serving as the main method used for this study. This will be used to explore system behaviour under multiple system configurations. Agent-based modelling effectively simulates interaction based on emergent behaviour following agent-level observations and subsequent actions (Nikolic & Kasmire, 2013). This is done in the light of the *Complex Adaptive Systems* paradigm, which perceives systems as the collection of self-organising autonomously operating entities (Nikolic & Kasmire, 2013; Waldrop, 1992). This implication works particularly well for conceptualising critical infrastructure affairs, as key actors operate based on their individual situational assessment, which is not fully rational when assess top-down (Liu et al., 2012; Rinaldi, Peerenboom, & Kelly, 2001; Teixeira et al., 2010; Van der Lei et al., 2010). The vast heterogeneity in ecosystem elements and properties discussed in section 1.2 can be implemented as agent-based modelling concepts to generate insight the effectiveness of defensive strategies.

### 1.4.4 Research sub-questions

To answer the main research question, a division is made into five separate sub-questions, each representing a different stage of research. This division is made to be able to generate more tangible answers to intermediate products. Together, the answers to these sub-questions will be synthesised into a coherent, substantial answer to the main research question.

Sub-question 1: *How does architectural complexity of critical infrastructure nodes within the cybersecurity ecosystem affect infrastructure operation?*

This first sub-question seeks to establish elements that contribute to the complexity of critical infrastructure systems. As stated in section 1.1 and in the literature review in Appendix A, the cyber-architecture of critical infrastructures opened the sector up to a large threat landscape. An answer to this sub-question would contribute to an improved understanding of the ecosystem and form the first step towards creating a simulation model. Identifying key assets and elements that comprise the ecosystem helps ensure that further specification is constrained properly.

Sub-question 2: *How do control mechanisms and cyber-threats secure or impede operation of critical infrastructures?*

The second sub-question follows up on the first sub-question and seeks to establish direct effects of control mechanisms and cyberattacks on critical infrastructures. These elements relate to mechanisms that directly affect effectiveness of cyberattacks. Answering this sub-question can help specify the process by which attacks take place and specifically how these relate to the elements discussed as part of sub-question 1.

Sub-question 3: *Which properties for attacker and defender behaviour aptly describe decision-making behaviour in the cybersecurity ecosystem of critical infrastructures?*

The third sub-question aims to establish a set of properties for attacker and defender behaviour in order to further specify interaction among entities within the cybersecurity ecosystem of critical infrastructures. Identifying the mechanisms that constrain and delineate interaction is crucial for conceptualising an agent-based model, as this method revolves around the notion of agent-level

decision-making. Modelling critical infrastructure requires extensive identification of these elements, as it ties in directly to the scope of analysis. Establishing decision-making models for cyberattacks is a focal point of cybersecurity research, and critical infrastructures are no exception to this (Brown et al., 2006; Ten, Manimaran, & Liu, 2010). A wide variety of academic literature exists on specific behavioural models for critical infrastructure cyberattacks, yet these are typically tailored to static, isolated infrastructures, as opposed to a dynamic interconnected ecosystem that will serve as the backbone of this modelling study (Ashok, Hahn, & Govindarasu, 2014; Baiardi et al., 2006; Cárdenas et al., 2011; Pasqualetti, Dorfler, & Bullo, 2013; Pawlick & Zhu, 2017; Rybnicek, Tjoa, & Poisel, 2014; Teixeira et al., 2010; Vuković, Sou, Dán, & Sandberg, 2012). The inclusion of these models in an ecosystem-level agent-based model built upon cyber-architectural elements could shed light on interactions within this ecosystem, as well as concepts central to interaction within the ecosystem.

Sub-question 4: *Which emergent behavioural patterns can be observed in interactions within the cybersecurity ecosystem for critical infrastructures?*

The fourth sub-question relates to emergent behaviour identified in simulation model outcomes. This sub-question requires the simulation model to be formalised and implemented, having verified and validated the model. Conducting exploration and experimentation results in behavioural tendencies of entities operating in the ecosystem. These behavioural tendencies can prove interesting, as they could provide insight into the effects of certain interventions. These tendencies can be used to assess the robustness of certain configurations for control mechanisms and could help understand the array of possible behaviour. This will be assessed by using various system configurations, with deviations in defensive strategies, as well as deviating between internal model states to ensure the exploratory model circumvents sensitivity.

Sub-question 5: *What can be learned about the effectiveness of defensive strategies with regards to robustness and resilience in the cybersecurity ecosystem for critical infrastructures?*

The fifth sub-question is similar to the fourth sub-question, as they both relate to outcomes from exploration and experimentation. This sub-question seeks to achieve statistical evaluation of system performance under certain configurations of scenario parameters, exploring the effectiveness of different defensive strategies. To achieve this, the data resulting from extensive simulation will be assessed to determine overall performance on key performance indicators, but also to establish possible leverage points that disrupt model outcomes in certain cases. The outcomes of this sub-questions will be comprise of evaluation of effectiveness of cyber-defensive mechanisms. Together with the fourth sub-question, this can help answer the last part of the main research question. Establishing recommendations for effective, coherent defensive strategies is the last requirement to answer the main research question, which should include substantiated findings regarding the interaction within the cybersecurity ecosystem and the sensitivity to mitigation strategies. The way elements from each sub-question will be used to answer the main research question is shown in Figure 1-2 below.

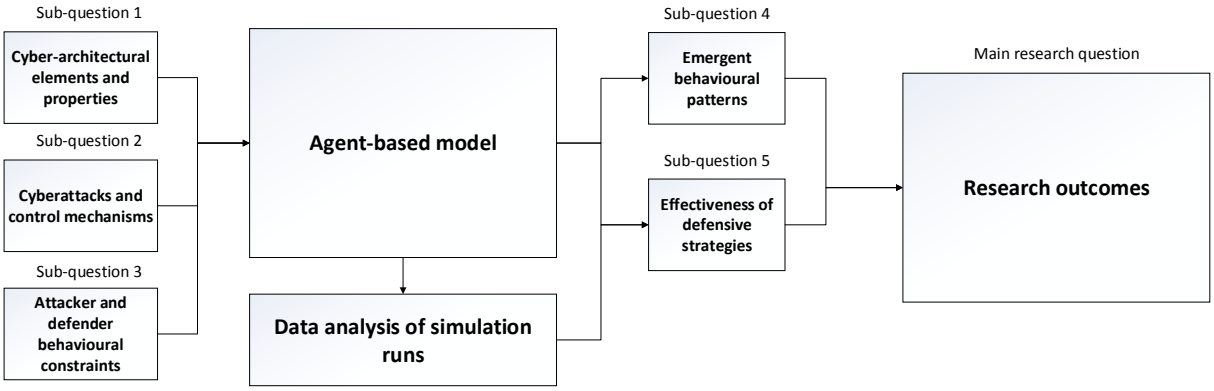


*Figure 1-2: Methods and data objects related to the agent-based model*

## 1.5 Thesis structure

In order to adequately answer the main research question, each of the six sub-questions will need to be answered. The application of the aforementioned five methods requires a clearly delineated research structure in which each step results in tangible answers or resources for further steps. The research will take place in seven steps, which are visualised in the Research Flow Diagram shown in Figure 1-3 below. For each major research phase, the corresponding steps in the agent-based modelling cycle by Nikolic, Van Dam, and Kasmire (2013) discussed in the third section are shown on the right side. The typical 10 stages of agent-based modelling by Nikolic et al. (2013) are represented by the formulated sub-questions: sub-questions 1, 2 and 3 relate to the system analysis phases, whereas sub-questions 4 and 5 relate to the relevance and validity of model findings and usage of the model. Together, the set of research questions provides insight required to establish an agent-based model and a clear direction to use the model to answer the main research question.

Methods to be applied during each research step are shown in green. Three major phases of research indicate the type of activities that will take place:

(1) Specification and conceptualising of the system-of-interest
(2) Formalisation and implementation of an agent-based model of the system-of-interest
(3) Exploring system behaviour under different configurations for defensive strategies

The distinction between sub-questions 1, 2 and 3 and on the other hand 4 and 5, as discussed in the third section, is represented in the Research Flow Diagram as well, as the first phase revolves around gathering the required information to develop a conceptual model and the second phase relates to the findings from model experimentation.

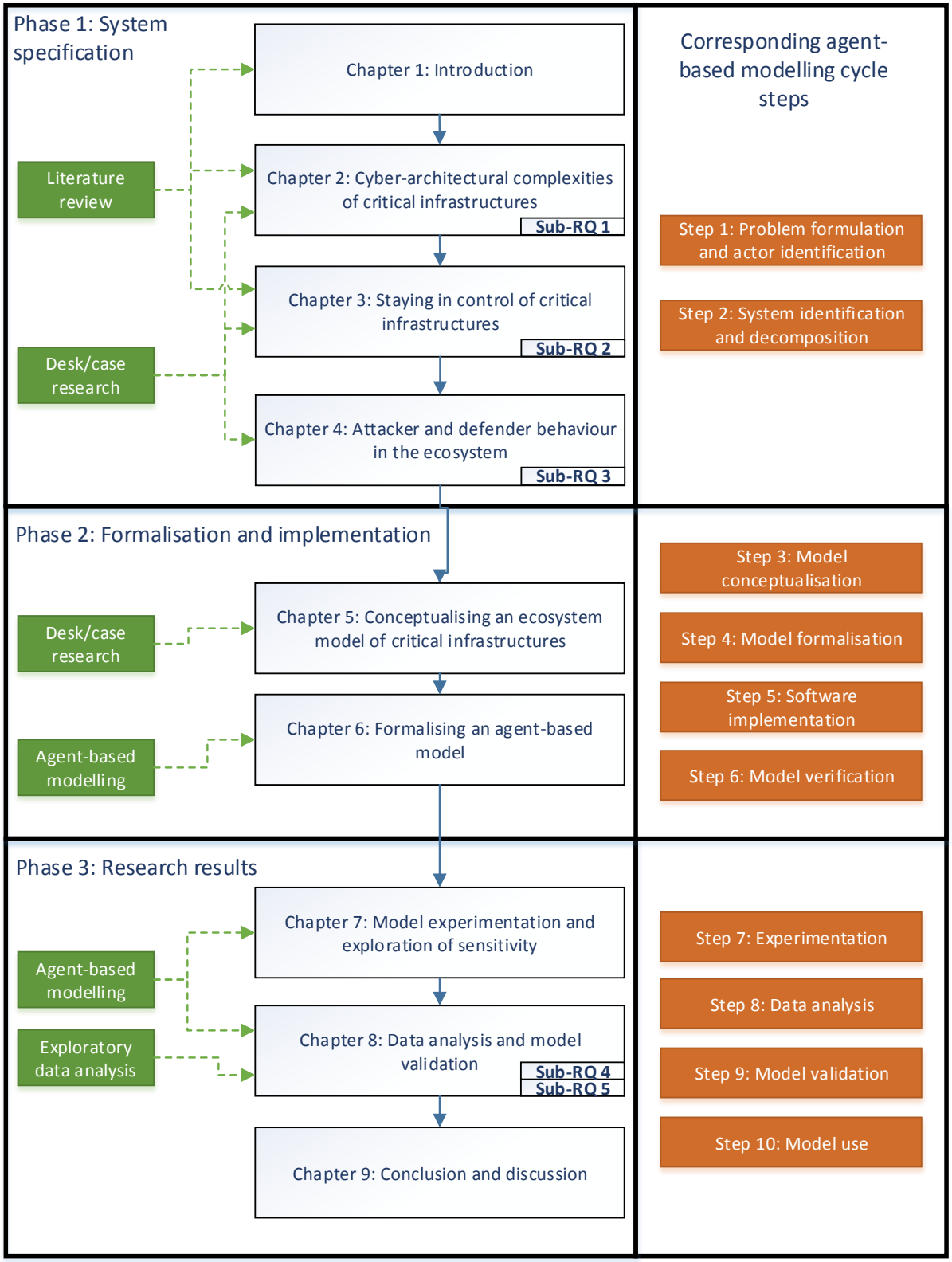


*Figure 1-3: Research Flow Diagram*

# 2 Cyber-architectural complexities of critical infrastructures

As stated in the first chapter, a critical infrastructure (CI) forms a complex system, characterised as a networks of distributed and interdependent subsystems with multiple distributed elements. Connecting additional subsystems allowed for critical infrastructures to carry out more tasks and increase their efficiency, but also opened systems up for additional threats. This chapter details the background of critical infrastructures, seeking to establish an answer to sub-question 1:

*How do critical infrastructure operators secure their operability against cyber-threats?*

This chapter expands on the conceptual overview defined in Figure 1-1 by focusing primarily on properties and concepts related to infrastructural nodes. This is highlighted in Figure 2-1 below. In order to answer sub-question 1, a background of cybersecurity for critical infrastructures is detailed.

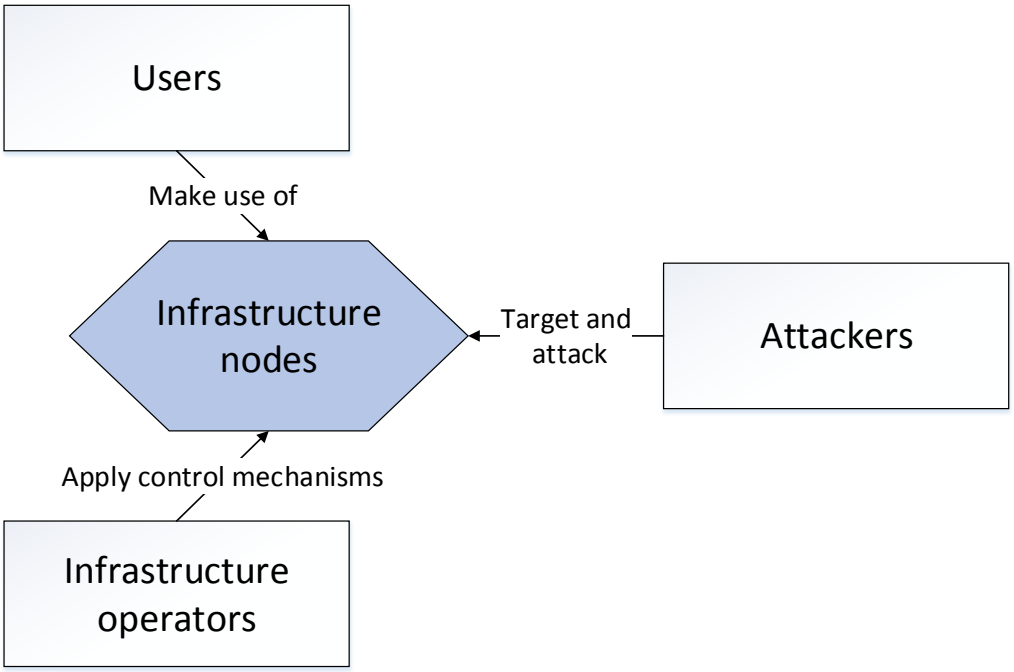


*Figure 2-1: Highlighted elements from the conceptual overview to be discussed in this chapter*

The first section provides insight into their complex architecture and how dependencies contribute to the threat landscape. The second section follows up on this by laying out the consequences of cyberattacks. The third section concludes this chapter by offering intermediate findings required to answer sub-question 1.

## 2.1 Architectural elements of critical infrastructures

In order to understand how cybersecurity issues arose ecosystem for critical infrastructures, their architecture must first be understood. This section decomposes the complex architecture of CI systems in order to establish a foundation for an eventual simulation model. First, the complex networked structure will be detailed. This is followed by a description of dependencies between infrastructure nodes. Subsequently, the heterogeneity between different infrastructural sectors is discussed. The resulting impediments from these three elements will be discussed afterwards.

### 2.1.1 Heterogeneity as a complicating factor

The core of cyber-risk in critical infrastructure systems is inherently rooted in the network architecture applied. Critical infrastructures often involve operation of Networked Control Systems (NCS), which rely on central systems to enforce control and supervision on networked subsystems

(Amin et al., 2013a). A key notion of nearly every critical infrastructure system is a *Supervisory Control And Data Acquisition* (SCADA) system: a central entity that monitors system operation across multiple linked subsystems (Miller & Rowe, 2012; Zhu, Joseph, & Sastry, 2011). More traditional, legacy systems integrated new system components in the form of sensors or other additional equipment in direct connection with SCADA systems. This corresponds with the transition towards digital communication within a hierarchical structure. To this date, many systems involving legacy components still deal with parts of hierarchical structures, which were originally not implemented with cybersecurity challenges in mind. Additionally, critical infrastructures that are directly involved with public health of many civilians adhere to a more direct and centralised structure, such as drink water management and flood protection systems (Abrams & Weiss, 2008; Amin et al., 2013a; Department of Homeland Security, 2015; Rasekh, Hassanzadeh, Mulchandani, Modi, & Banks, 2016). This allows those systems to enforce security elements on the crucial, central system to prevent failures in the most critical part of the system. The most important consequence from this, however, is that an increased reliance on additional equipment was not subjected to the same security standards as SCADA systems.

Modern critical infrastructure systems integrate a wide variety of different subsystems. This shift towards decentralised and distributed networks of heterogeneous system components made it possible to collect more information within the network. Instead of relying on a single entity to assert control, control over system components is distributed among nodes. These nodes are represented by certain types of system components, such as task-specific control systems, physical sensors and communication links, each subject to different standards and protocols (Amin et al., 2013a; Khurana et al., 2010). These systems ensure that an easy path to control nodes does not diminish the integrity of the entire network. On one hand, additional information made possible by this wide array of information gathering tools significantly increased the effectiveness and possibilities for critical infrastructures. On the other hand, the transcension of critical infrastructures beyond the physical domain has also opened up the system for undesired access from the cyber domain.

Using digital communication to assert control over parts of networked control systems such as CIs offers significant benefits, despite unforeseen consequences. What was originally not expected was the way by which attackers could eventually access crucial system components more easily. Traditional, hierarchy-based infrastructures never accounted for the possibility of attackers affecting SCADA systems by targeting distributed sensors. Attackers could target vital national infrastructure assets through small-scale sensor disruption or intrusion causing enhanced consequences (Department of Homeland Security, 2015). The connectivity of critical infrastructures proves to enable attack vectors to be abused by malicious actors. Intrusions in critical infrastructure systems that result in substantial consequences can be conducted more easily (Lee et al., 2016; Romanosky & Goldman, 2016). This resulted in a stark increase in impact and frequency of cyberattacks on civilian targets, highlighted by large-scale attacks on the Ukrainian electricity grid in 2014 or the Stuxnet infection of Iranian nuclear facilities (Farwell & Rohozinski, 2011; Karnouskos, 2011; Lee et al., 2016; Liang et al., 2017).

Whether control systems for critical infrastructures were designed around legacy SCADA systems or modern networked systems, the challenge faced is the same. Both the increased burden on critical infrastructures and widespread availability of information required for infrastructure operation require an agile yet robust system. While these aims seem contradictory, the challenge rests in juxtaposing critical infrastructure between robust system operation and adapting to intrusions or disruptions. Networked control systems should be able to cope with partial unavailability of network elements. Disruptions within a certain section should ideally not extend beyond that section. On the

other hand, attackers should not be able to breach SCADA nodes by gaining access to connected system nodes.

Another complicating factor is rooted in the type of subsystems being used to gather information, communicate and assert control. In many such cases, commercial off-the-shelf software and hardware is integrated in order to save resources (Ericsson, 2010; Hull, Khurana, Markham, & Staggs, 2012; Pederson, Dudenhoeffer, Hartley, & Permann, 2006). Conversely, usage of COTS products also impedes security practices, as it might bring vulnerabilities into an otherwise secure environment. Trust is shifted towards third party producers of COTS software and sensors. Most importantly, the added heterogeneity prevents infrastructures from updating security elements coherently. The more different sources of software and hardware present, the more problems will be experienced in applying patches and making minor modifications to resolve issues.

This sub-section defined several elements that are classified as heterogeneous properties of the cyber-architecture of critical infrastructures. However, these elements all relate to a degree of internal operability: they determine how robust and resilient an infrastructure can ultimately be. On an ecosystem level, the desired elements all relate to how these can impede the functioning of a CI system. While specific node-related elements might require further specification for security assessment of individual nodes, this adds little value for an ecosystem model (Eid & Rosato, 2016; Rinaldi, 2004). With focal point established, it is important to realise that analysis of an interdependent and interconnected ecosystem might not lead to tangible policy options for single infrastructure nodes. In fact, it is essentially impossible to devise a single monitoring system for all infrastructure elements. On the other hand, this scope could prove useful for exploring possible ecosystem-level interaction. Shared strategies in terms of dealing with these elements are not impossible.

### 2.1.2 Dependencies and cascading failures

The second concept crucial to the identity of critical infrastructures is the presence of dependencies and interdependencies among networked systems. While slightly touched upon in the previous sub-section, dependencies are one of the key features that distinguish critical infrastructures from regular cyber systems, and therefore require further specification. Dependencies between critical infrastructure systems imply direct effects from hindered operation in one node to another. Many incidents in critical infrastructure systems involved failures in sequential nodes, leading to a chain of events causing increasingly higher damage sustained. Dependencies can also work both ways, where two systems are interdependently affected. Failures caused by dependent nodes are known as cascading failures. Cascading failures as a concept is widely discussed with regards to industrial control systems, where direct industrial processes affect each other. This degree of dependency can occur between two nodes within the same infrastructure as well as between two infrastructures.

The first type, dependencies and interdependencies among infrastructure network nodes, directly affects the capabilities of individual nodes. These dependencies will be referred to as *first order dependencies* (Setola & Theocharidou, 2016). First order dependencies involve direct connection between two nodes, typically involving sequential reliance on information or physical control on another node. For example, power grid load balancing systems are directly affected by unavailable information flow from different sensor nodes at other locations. This type of dependency is encountered in every networked control system by pure definition. Further connections between nodes create multiple-order dependencies, indicating how easily small disturbances can spread across a large network.

The second type of dependencies involves dependent infrastructure nodes across different networks. Cross-sectorial dependencies are different from regular first-order dependencies in the sense that the two involved nodes are part of different systems. Cross-sectorial dependencies are not a new concept, as these already existed in traditional, physical infrastructures. A 2001 train derailment in the United States resulted in widespread damage beyond what was originally expected (Pederson et al., 2006). Damage to a train tunnel caused physical damage to a water main, which in turn led to a flooding that disrupted the local power network. In the light of modern, cyber-physical infrastructures these dependencies are a lot more subtle than a chain of physically damaging events. In many cases, nodes require basic operation of associated nodes, and failures within one network quickly spread across that very network. This also holds up for dependencies between networks. An example of such a cross-sectorial dependency is water purification systems relying on a stable electrical power grid or the electrical grid relying on information from decentralised power generation (Department of Homeland Security, 2015; Pederson et al., 2006). A conceptual overview of such cross-sectorial dependencies, created by (Pederson et al., 2006), is shown in Figure 2-2 below.

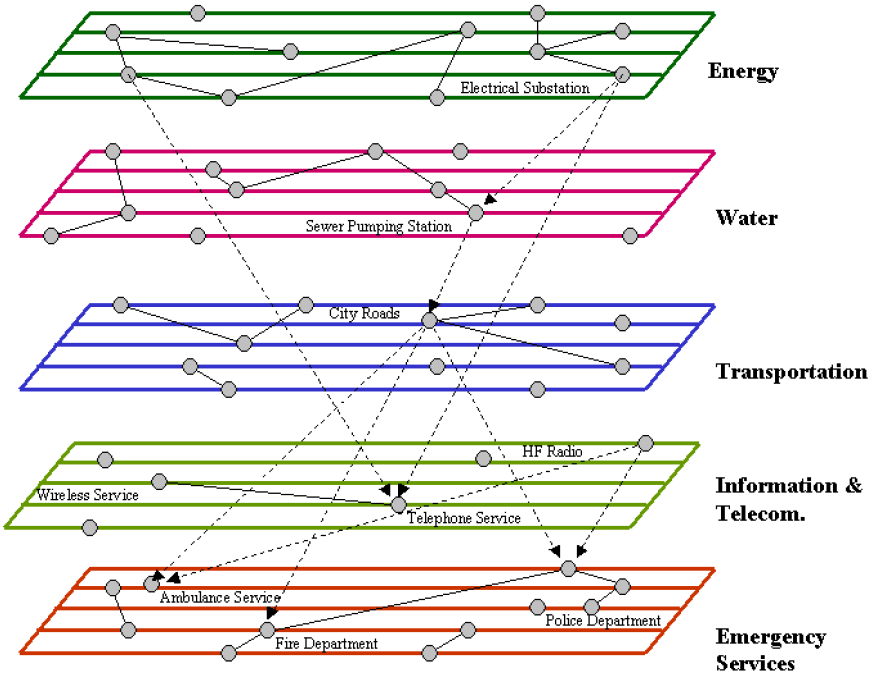


*Figure 2-2: Node-level dependencies and cross-sectorial dependencies represented as a graph, by Pederson et al. (2006)*

Categorising dependencies can help understand the processes that lead to loss events, but do not necessarily represent the consequences involved with these events. Depending on the weight given to dependencies, these have a likelihood of causing a cascading failure in dependent nodes. A model for this interaction was proposed by Setola and Theocharidou (2016) and is shown in Figure 2-3. This model incorporates dependencies as weighted edges between nodes in a network graph, which corresponds with the description given of dependencies and CI cyber-architecture in this section. These weightings or influence coefficients enable modellers to create scenarios for cascading failures. Scenarios based on dependencies, external pressure and possible attacks can help yield insight in how risk spreads under certain system configurations. Crucially, the effect of these dependencies affect the same core notion of operability within a node. An ecosystem of infrastructures therefore includes infrastructure nodes that incorporate dependencies that affect the level of operability, which is different among nodes. For modelling purposes, regular dependencies

and cross-sectorial dependencies are not inherently different. This results in the following, standardised assumption: dependency weightings directly affect the state of operation for dependent nodes based on the state of operation for the other end.

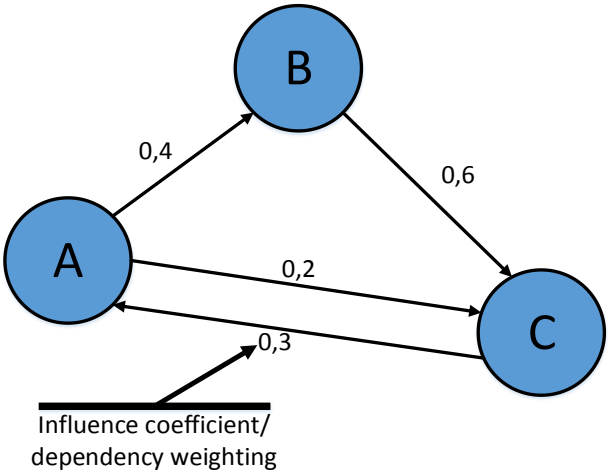


*Figure 2-3: Infrastructure node dependency scenario, by Setola and Theocharidou (2016)*

### 2.1.3 Cross-sectorial differences in topology

The third key aspect concerning the complexity of critical infrastructure systems revolves around heterogeneity between different infrastructures. Whereas the first sub-section detailed the general complexity of networked systems and the second sub-section went into detail as to how system elements affect one another, it is also important to denote the effects of different networked systems and different data elements. Aggregating concepts into a coherent ecosystem should account for differences that the topology of a network might make. As stated in sub-section 2.1.1, there is a general divide between traditional SCADA-based systems and contemporary, distributed networks.

The hierarchy of nodes within any infrastructure is directly associated with the type and weighting of dependencies found within the network. CIs that involve concentrated, physical infrastructure, such as flood protection systems, are more likely to depend more heavily on geographical events related to other nodes in the network than networked systems that rely on physical and cyber characteristics of nodes in the network. More importantly however, the hierarchy of nodes within infrastructural sectors cannot be detached from their history and development over time. Since sections 2.1.1 and 2.1.2 established that heterogeneous infrastructure nodes operate identically on the level of abstraction required to assess defensive strategies, this is not problematic. The model should integrate these factors without necessarily specifying matters too similar to individual infrastructures. The model should be generalisable for generic properties for CI systems, and remain extensible to include different network topologies.

As mentioned in sub-section 2.1.1, infrastructures that carry a heavy physical burden on society are more likely to base their network topology around a centralised SCADA system. Any developments in terms of new sensors or other system components are moulded around this central system. While this calls for more specific development, it makes sure that interoperability can be maintained. The legacy nature of these networks implies that upgrading happens at a slow pace, if at all. Security of distributed components was often thought to not be an issue so long as the SCADA system was secured well, but the degree to which dependencies influence system operation can still result in cascading failures. For lower numbers of control nodes in a network, the relative dependency on

each associated node increases. In many such cases the problem is related to a weakest link that can significantly hinder operation.

In modern CI systems, the challenge shifts towards the other direction. More modern architecture applied to newer emergent critical infrastructures make use of more decentralised network nodes and distributed authority. An example of this type of critical infrastructure is the smart electricity grid (Ericsson, 2010; Wei, Morris, Reaves, & Richey, 2010). Sensors and communication components are used to measure contemporary electrical loads at many different locations, including generation, transmission, usage and regeneration (Khurana et al., 2010; Yan, Qian, Sharif, & Tipper, 2012). The problem with decentralising critical infrastructures is the lack of control that can be asserted over the network as a whole. Differences in standards, protocols and software packages used greatly affect the amount of information available within the system. For operators of individual subsystems it then becomes incredibly difficult to assess the current situation in terms of risk exposure.

Since the types of dependencies have already been clarified, the remaining key to the puzzle is how connectivity models correspond with the hierarchy of CI systems. Centralised control systems resemble a star topology based on a single control node. Outer nodes can be dependent among each other, but essentially all outer nodes are connected to the same central SCADA node. Decentralised systems are formed by a less hierarchical network of individual nodes and smaller subsystems. From a connectivity perspective, the model an infrastructure is based on determines the attack surface for possible attackers, as intruders can access subsequent nodes more easily (Alcaraz & Lopez, 2013). Besides, enforcing dependency on other nodes as opposed to duplicating information or resources could lead to cascading failures (Svendsen & Wolthusen, 2007). On the other hand, applying a greater number of inner connections can help avoid having large portions of the infrastructures being cut off completely (Buttyan, Gessner, Hessler, & Langendoerfer, 2010). This does result in a greater level of dependencies, and might increase the risk of cascading failures.

In order to translate the concept of critical infrastructures into a simulation model, the topology of network nodes needs to be clarified. Link topology represents the arrangement of nodes in the system. This is based on inner connectivity and dependencies. Connected nodes are part of a physically or logically linked whole, whereas dependent nodes are functionally linked (Setola & Theocharidou, 2016). Dependencies affect the level of operation directly, whereas connections open the system up for possible external intrusions (Liang et al., 2017). The ecosystem model should therefore include both inner connectivity as well as dependencies.

### 2.1.4 Complication of ecosystem-level control

The issues discussed in the previous three sub-sections provide insight in how critical infrastructures operate. It was identified that CI systems encounter issues rooted in the usage of complex heterogeneous components, dependencies among network nodes and heterogeneity between separate infrastructures. These three elements form the most prominent issue in securing critical infrastructures: a lack of agility enabled by current architecture. This does not provide sufficient insight into how process of controlling the system as a whole is affected. Instead, urgent development of architecture-based security solutions are necessary (Khurana et al., 2010).

The desire for cybersecurity in CI systems has urged many researchers to study the effects of CI failures induced by cyberattacks. In many cases, a specific case raised urgency for analysis of a specific infrastructural sector, the most prominent being the scramble for research into cyber-resilience of power grids following Stuxnet and the 2015 Ukraine blackout (Fairley, 2016; Farwell & Rohozinski, 2011). In many cases, such an approach fails to sufficiently emphasise dependencies and interdependencies. According to Clark et al. (2017), the challenge in securing individual CIs requires a

coherent cybersecurity culture. Furthermore, they describe the desire for a secured network design to be implemented across CI systems. This is underlined by both Neuman (2009) and Karnouskos (2011), who urge the need for cybersecurity challenges to be a central element throughout the design process.

Given the lack of coherent defensive strategies to effectively thwart cyberattacks, CI systems require network-level analysis. Looking at individual infrastructures in isolation might yield more direct and interpretable results, but fails to represent real-world challenges, which are deeply rooted in cross-sectorial interdependencies (Rinaldi, 2004). Representing a network of infrastructure nodes built around a set of weighted dependencies can be crucial to determining the spread of risk and accountability throughout a control system (Vuković et al., 2012). It is therefore crucial to explore different network designs as a crucial element of cybersecurity. By establishing a set of shared vulnerabilities and controls among CI systems, cybersecurity can be explored on an ecosystem level (Clark et al., 2017; Pederson et al., 2006; Ten, Liu, & Manimaran, 2008). These shared vulnerabilities and controls are based on the architectural concepts discussed throughout this section and will be further detailed in chapter 3. While this approach might yield generalisable network-level results, it raises another question in terms of authority and manageability. The multitude of heterogeneous system components, including specifically designed sensitive systems as well as commercial off-the-shelf systems, makes it difficult to accurately prescribe a path forward (Department of Homeland Security, 2015; Ericsson, 2010; Pederson et al., 2006). A security-by-design network model that effectively mitigates cyber-risk does not necessarily translate into actions that current infrastructures can apply. On the other hand, it enables coherent security policies that can be integrated and enforced top-down, to ensure similar actions are being taken that have proven to actively reduce risk exposure.

## 2.2 Consequences of cyberattacks

As stated in chapter 1, the consequences of attack-induced failures in CIs are immense. Wrongful identification of attacks could lead to incorrect defensive actions, which might hurt system operation without an actual attack taking place (Department of Homeland Security, 2015). This section lays out a framework for the types of damage sustained in CI systems and how consequences are dealt with. Stamp, Dillinger, Young, and DePoy (2003) lay out a framework of three forms of impact for critical infrastructures: *physical* impact, *economic* impact and *social* impact. Since social impact for CIs is a rather tenuous concept that is difficult to measure, it is left out of the scope of this study. For private organisations and companies it makes more sense to include elements such as reputational losses, but this does not hold up as well for critical infrastructures. Reputational damage sustained to or public confidence are direct consequences from the first two types and will therefore be complimentary to those types. To illustrate consequences in both the physical and economic domains, the pathways detailed in Department of Homeland Security (2015) will be used.

### 2.2.1 Physical damage

Physical damage is identified as the direct material loss in key assets. This includes damage to property, loss of life and environmental assets (Stamp et al., 2003). The two most straightforward physical attack vectors provided by the DHS report involved disruptions in the transportation and water management sectors.

Smart city developments have added many functionalities to vehicles, automating tasks that would originally be left to humans driving the vehicle. Researchers have already exposed several vulnerabilities in vehicles, allowing malicious actors to remotely control crucial vehicle functions. Large-scale attacks on multiple vehicles could lead to multiple serious traffic accident, leading to

significant damage to key infrastructure and likely also loss of life. Other possibilities include tampering with train signals or disrupting road traffic signals. Disruption of these services can directly result in traffic accidents that pose serious risk to public health.

Developments in the water management sector allowed for wide-area monitoring of many different variables. Water management systems can keep track of the concentration of chemicals or substances in drinking water to detect contaminations, and flood protection systems are capable of controlling physical flood barriers remotely to responsively act on certain scenarios. Cyberattacks on these systems have potential to cause significant damage to public health. One possible attack vector laid out in the DHS report, corroborated by Stamp et al. (2003), makes use of the remote accessibility of wastewater facilities to cause a flow of wastewater into drinking water, endangering public health and damaging the environment. A more direct attack vector is to infiltrate flood protection systems, which can directly lead to mass flooding in busy residential districts. Disrupting storm barriers in emergency situations cold possibly lead to physical harm of thousands of residents.

### 2.2.2 Economic disruption

Economic damage is often considered a second-order consequence of physical disruptions (Stamp et al., 2003). Economic impact is defined as the consequences of physical impacts on system operations, which transcend beyond the scope of the original asset.

Whereas the physical impact of cyberattacks on CI systems is often directly noticeable, economic disruption as a result of those attacks is almost always inevitable. Besides direct damage to the environment, unavailability of infrastructures has substantial effects on the economy. Electricity blackouts lead to a loss of productivity for many companies affected, whereas traffic disruptions would prevent many employees from getting to work. An unexpected closure of a New York bridge in 2013 caused over $7 million in economic damages to the local economy, in addition to impeding emergency services (Department of Homeland Security, 2015).

Economic disruption is almost always a consequence of more subtle attacks, oriented around financial gain for attackers. Hijacking smart electricity meters costs the U.S. economy up to $6 billion annually (McLaughlin, Podkuiko, & McDaniel, 2009). Such attacks extend beyond stealing assets, as smart meters also allow for attackers to cut power to consumers or access home automation systems. While generally considered a second-order consequence from cyberattacks, economic disruption can very well be the main consequence of a cyberattack.

### 2.2.3 Resilience

Defending critical infrastructure around outside threats (including cyber-threats) requires clear identification of the type of damage to be expected. A resilient infrastructure is one that sustains little expected damage. To this end, a military approach discussed by Brown et al. (2006) includes four evaluations for an asset: the *criticality* of an asset, the *vulnerability* of an asset, the *reconstitutability* for losses and the *threat* likelihood. These evaluations seek to establish the very core of an enumerable degree of damage. However, the applicability of a military damage assessment is questionable. The planning involved for defending military assets often involves more complicated testing on a closed-off environment (Brown et al., 2006). The complicated testing involved for military-grade cyber-resilience involves tests for collateral damage and tactical positioning of certain assets (Hare & Goldstein, 2010; Romanosky & Goldman, 2016; Thompson, Morris-King, & Harang, 2016). Civilian critical infrastructures are much more open in character and assets are less expendable. As a result, worst-case analysis is required to get to the root of threats at any point in time. This approach allows researchers to maintain a multilevel definition of damage for attacker-defender models (Brown et al., 2006). In the light of dependencies and architectural

elements discussed in section 2.1, this lets a simulation model establish a coherent collective of all core concepts. Attacker-defenders models will be discussed in more detail in chapter 4.

Despite the differences, the military four-step evaluation is agile to work in the context of civilian assets as well. While the strategic element of allocating resources based on likely targets is not applicable to this study, it is important to recognise the importance of assumptions made for both attacks and assets. For an ecosystem-level model that yields any meaningful interaction, attacks should be sufficiently present. The consequences should therefore follow standardised formalisms. The results that might arise from this assumption should be discussed in the light of this assumption. In order to deal with a dynamic ecosystem, attacks are modelled as sufficiently surmountable to enable tracing attributable elements of decision-making.

## 2.3 Intermediate findings

Throughout this chapter, elements related to the cyber-architectural structure of critical infrastructures were discussed in relation to risk management. The first section details how the decentralisation and distribution of infrastructural elements allowed for greater synergy within the ecosystem, while also enabling many pathways for cyberattackers. It was found that both first-order dependencies and cross-sectorial dependencies greatly affect the ability for individual system elements to operate, and that interoperability issues can quickly cause cascading failures. These elements complicate the options for security modifications, as networks each use different elements for different tasks, highlighting the need for coherent security strategies. The second section discussed the degree and types of damage that CI systems can be exposed to as a result of cyberattacks. The main consequences are physical and economic damage that can lead to further widespread cascading failures. The resilience of an infrastructure asset at any point in time is tied to its criticality and reconstitutability. Attacks encountered in the ecosystem should enable dynamic decision-making and are therefore assumed to be surmountable. These concepts are all added to the basic conceptual overview shown in Figure 1-1, resulting in the expanded model shown in Figure 2-4.

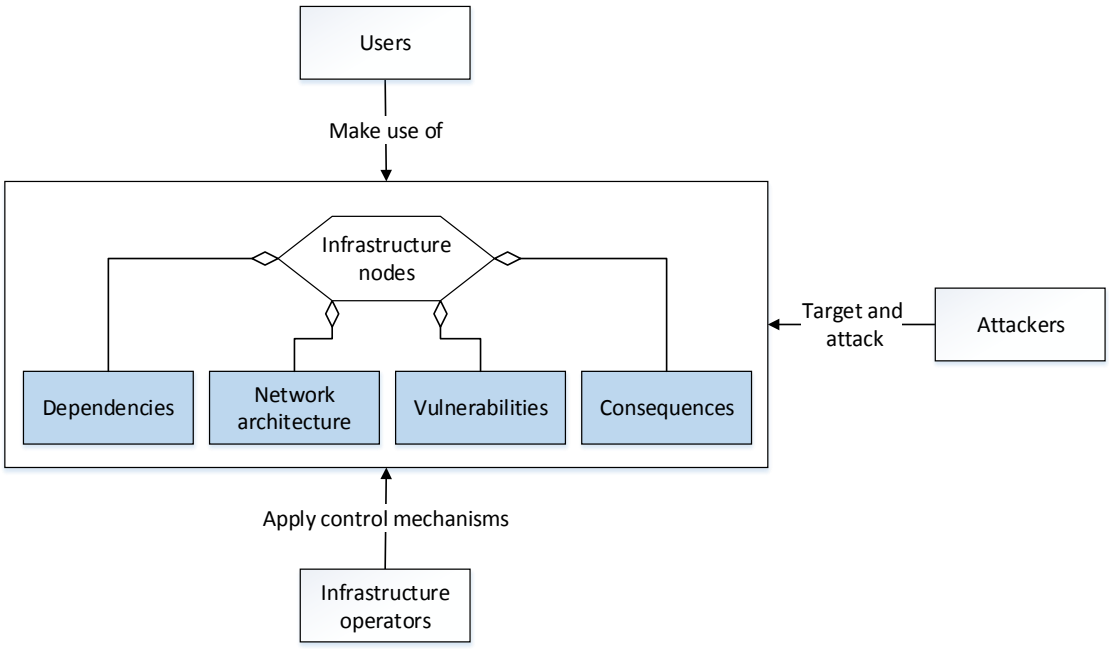


*Figure 2-4: Expanded conceptual model with new additions highlighted in blue*

# 3 Staying in control of critical infrastructures

This chapter details the cybersecurity elements of critical infrastructure systems in the light of attack and defence events. The elements discussed throughout this chapter build on the foundation laid in chapter 2, extending the conceptual aggregation from a cybersecurity. This chapter seeks to build evidence required to answer the second sub-question:

*How do control mechanisms and cyber-threats secure or impede operation of critical infrastructures?*

This chapter relates to offensive and defensive mechanisms involved in securing critical infrastructures. To accomplish this, the background of threats and defensive strategies will be established. The additions to the conceptual model shown in Figure 1-1 are therefore related to attackers and infrastructure operators. This is highlighted in Figure 3-1 below.

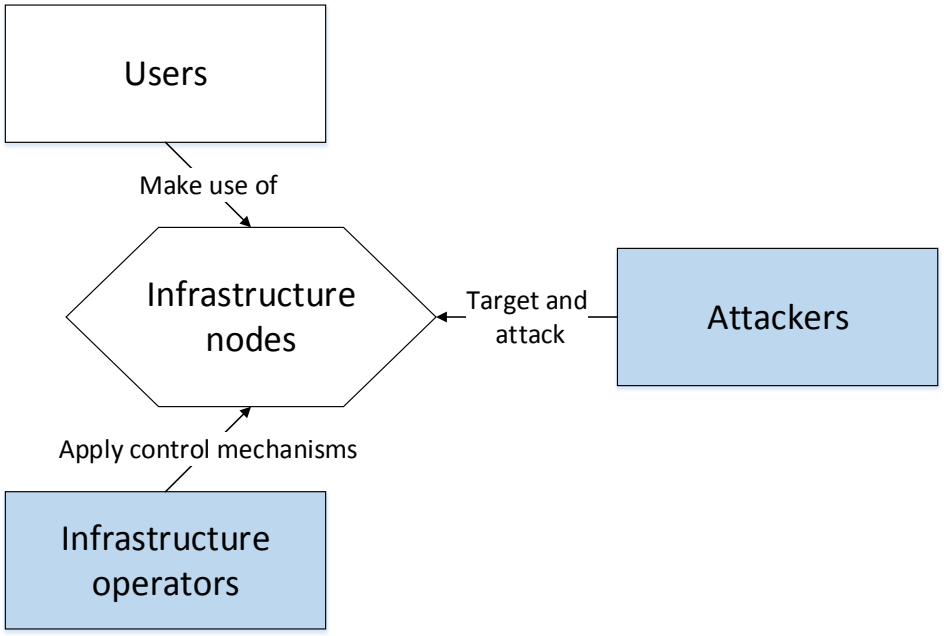


*Figure 3-1: Highlighted elements from the conceptual model to be discussed in this chapter*

To accomplish this, the first section will provide a definition of cybersecurity concepts in the light of critical infrastructures. These definitions extend the previously discussed architectural elements of CI systems and will help build towards a representative simulation model. The second section details the types of attacks experienced within CI systems and networked control systems. The third section follows up on these attacks to provide a set of control mechanisms used to thwart and mitigate cyberattacks. The fourth section wraps up this chapter by offering intermediate findings.

## 3.1 A cybersecurity perspective on CI systems

Security challenges for cyber-physical control systems rely on two key definitions that will be provided in this section. The first step is to define what cybersecurity entails within the context of this study. The second step is to define cyber-risk and how this concept relates to the ecosystem of critical infrastructures.

### 3.1.1 Defining cybersecurity

Modern critical infrastructures are the product of decades of technological advancement. Traditional control theory approaches could only ensure a very basic level of protection against cyber-threats (Fairley, 2016; Sandberg et al., 2015). The inability to account for an open flow of information within infrastructure networks is a primary driving force behind the degree of risk (Department of

Homeland Security, 2015). On the other hand, the issue at hand does not correspond with an information security problem either (Amin et al., 2013a). Information security revolves around three concepts as denoted by Van den Berg et al. (2014) and by Rasekh et al. (2016):

1. Confidentiality of information
2. Integrity of information
3. Availability of information

These concepts cover the essence of cybersecurity, but fail to address the capabilities of malicious, intelligent attackers (Amin et al., 2013a). Cybersecurity involves protecting assets that enable cyber-activities against intrusions originating from malicious actors, with particular focus placed on emergent socio-technical interaction (Van den Berg et al., 2014). Essentially, cybersecurity diverges from information security, which is primarily focused around securing specific data, towards securing IT-enabled activities. As stated in chapter 1, the emergence of cyber-threats against critical infrastructures originates from a shift towards IT-enabled activities. The severity of consequences from cyberattacks on critical infrastructures create an urgency for 'perfect' security models. In order to improve resilience of cyber-physical critical infrastructures, a cybersecurity approach is necessary.

The addition of physicality further stresses the need for a more tailored approach to critical infrastructures. The prominent presence of physical elements in CI systems means that a neither purely physical control nor purely cybersecurity fully covers resilience (Sandberg et al., 2015). A fitting approach therefore incorporates elements from both cyber and physical domains. This relates to interdependencies transcending beyond an isolated domain, as well as the degree of materiality resulting from cyberattacks (Aradau, 2010). An IT-focused security approach, while capable of accurately mapping properties of IT-specific assets, can simply not account for current threats to IT-enabled cyber-activities for CI systems (Sandberg et al., 2015).

### 3.1.2 Defining cyber-physical risk

Cybersecurity analysis typically involves securing assets against a concept defined as *risk*. A commonly used definition of risk based on the *Factor Analysis of Information Risk* (FAIR) framework by Jones (2006), as it provides extensible and adaptable concepts. The definition of risk rests among four concepts according to FAIR:

1. An *asset* is an entity that supports cyber-activities, where actions taken against this asset cause a form of loss.
2. A *threat* is an entity that acts against the asset to cause harm
3. *Vulnerability* is the degree by which the attacking capabilities of threats exceed the defensive capabilities of assets
4. *Risk* is the resulting combination of the probable frequency of loss events and the probable magnitude of loss events (Jones, 2006).

FAIR, as the name of the framework implies, revolves around identifying information security elements. These concepts are tailored around the direct internal consequences of cyberattacks. The architectural complexity of critical infrastructure networks makes the definition problematic, since dependencies introduce unexpected consequences for attacks on assets. This raises questions about the applicability of traditional risk analysis for critical infrastructures. Risk within the context of this study represents the current extent of consequences resulting from loss events. Therefore, it is best to avoid the term 'risk' and instead attribute related concepts to a degree of operability. A key distinction is made to differentiate between a quantifiable optimisation problem and the complex emergent behaviour that is to be explored through this study. The former could be solved by

identifying a degree of tolerated or accepted risk to determine optimal defensive strategies (Van den Berg et al., 2014). The latter is typical for CI systems, as there is simply no telling how vast the consequences of attacks will turn out to be, shifting the focus of risk analysis. The severity of consequences for critical infrastructures imply that such analysis is simply not feasible. The metrics used to define risk extend beyond the here and now, as risk enumerates the expected losses over a set time period. Within this study, risk is translated to a degree of infrastructure node operation.

However, the concepts of FAIR can be adapted to create a coherent framework that fits the scope of this study. The consequences of attack-induced failures for critical infrastructures are substantially larger than typical loss events described in Jones (2006), as successful attacks are somewhat rare occurrences that result in more significant losses (Department of Homeland Security, 2015). Within FAIR, the process that determines the likelihood of success for a single attack is based on a relatively simple subtraction of the defensive control strength from the attacking capabilities. For critical infrastructure systems, the sophistication of control mechanisms, cyberattacks and dependencies among assets forms a problem for this definition. Cybersecurity elements for critical infrastructures should therefore follow the susceptibility of control mechanisms to attacks in general. This avoids any excess specification, where core interaction in the ecosystem becomes too specific for generalisability. Vulnerability in the context of CI systems is an emergent property of complex interaction. This study is built around the mechanisms by which individual intrusion attempts affect system operation and how infrastructure operators attempt to assert control.

The resulting concept resembling cyber-risk in the system is thus the product of complex emergent interaction affecting operability of infrastructures. Defensive decisions rely on time-sensitive information on active threats and a subsequent perception of the active threat landscape. The degree of perceived operability of infrastructure nodes is subject to limited rationality, as mistakes can be made in assessing active threats (Teixeira et al., 2010). The processes by which they respond or employ control mechanisms will be discussed in section 3.3, and the limitations to rationality are further discussed in chapter 4. The degree of cyber-risk as present in academic literature is rather inapplicable for critical infrastructures, the concepts of which are primarily related to instances of attacks occurring. This indicates a shift in the scope of this study towards modelling active intrusions and defensive decisions made to mitigate these intrusions, as opposed to modelling a developing degree of security based on investment decisions. The model by extent incorporates worst-case analysis as the foundation for security assessment, as this implies the relevant degree of security that critical infrastructures should adhere to (Brown et al., 2006).

## 3.2 Taxonomy of cyberattacks

Securing CI systems relies on understanding all facets of the surrounding cybersecurity ecosystem. An important element of this ecosystem is the type of cyberattacks that take place. Understanding cyberattacks requires identification of both the types of attackers and their motivations as well as identifying the types of attacks and their attacks.

### 3.2.1 Types of attackers

The first element of cyberattacks to be discussed is the type of attacker involved. Attacker base their decisions on the information that is available to them and their preference for attack outcomes. Cybersecurity literature discusses many different types or profiles of attackers, yet critical infrastructures only involve a highly specific subset of those. The highly critical nature of CI systems makes them an interesting target for advanced persistent threats (APTs), and less interesting for typical attacker types (Pawlick & Zhu, 2017). Security incidents for CI systems involve highly specific intrusions, whereas regular cybercrime is only a small drop in the pond. The primary threats involved

in the ecosystem are therefore considered exclusively APTs, as the vast majority of possible cyberattackers do not have the skills or resources to conduct sufficiently powerful attacks (Herzog, 2011). However, this does not mean that the system is closed off for smaller infractions, as smaller external security threats can influence the system (Alcaraz & Lopez, 2013; Department of Homeland Security, 2015; Li et al., 2012). Three types of APTs are considered, along with their preferences for attacker utility as well as their capabilities. These three APTs are based on Clark et al. (2017), with government-sponsored entities grouped together. These types corroborate Rasekh et al. (2016), taking into account that internal threats do not follow similar targeting procedures.

*Foreign adversaries* are powerful, state-sponsored entities who seek to inflict large-scale damage to targeted infrastructures. These actors primarily engage in disruptive and destructive attacks, looking to harm foreign economies (Rasekh et al., 2016). Foreign adversaries typically do not seek to inflict damage to individuals, as that can be perceived as an act of war (Romanosky & Goldman, 2016). Their capabilities are immense and often make use of sophisticated worms, capable of intruding targeted sections or spreading through entire networks. Examples of these actors in recent years have been the sophisticated Stuxnet worm believed to have been an effort of international cyber warfare or the targeting of Georgian communication networks in 2008 (Farwell & Rohozinski, 2011; Karnouskos, 2011). The desired effects for foreign adversaries are to achieve relative gains compared to their target. In some cases, foreign adversaries might attempt to defraud assets for financial gain, although their motivations are primarily disruptive and destructive.

*Cyberterrorists* are malicious actors motivated to instil fear by causing large-scale damage and loss of life (Rasekh et al., 2016). Resilience against terrorist threats to critical infrastructures has grown to be an increasingly important goal for national defence (Moteff et al., 2003; Shea, 2004). Terrorists are capable of executing potent attacks, hoping to cause significant physical damage to targets. The primary motivations for cyberterrorists are rested in causing destructive and deadly consequences. Cyberterrorists make use of the open nature of CI systems to further a political or ideological goal, while secondary effects typically impact local economies and trust in public office (Moteff et al., 2003).

The third and last type of actors are *cybercriminals*. Cybercriminals make use of disruptive malware as a means of defrauding infrastructures or attend to other means for blackmail (Perakslis, 2014). Cybercriminals typically possess fewer resources than the other types of threats, of can still potentially cause widespread damage. An infrastructural sector that has grown increasingly exposed to such threats is healthcare infrastructure, with multiple recorded instances of data breaches leading to identity theft (Kruse, Frederick, Jacobson, & Monticone, 2017; Luna, Rhine, Myhra, Sullivan, & Kruse, 2016).

### 3.2.2 Types of attacks

The second key element required to understand cyberattacks is to establish a classification scheme for the possible types of cyberattacks. The classification scheme to be used is adapted from Miller and Rowe (2012). Their scheme, based on a survey of attacks on CI systems, incorporates four factors for classification, each with multiple facets:

1. Source sectors
2. Method of operation
3. Impact
4. Target sectors

This scheme enabled the authors to denote a wide variety of incidents. However, this classification is rather broad, as it served a primarily normative purpose. A simulation model based around the architectural complexity of interdependent infrastructure nodes requires a more concise classification scheme. The target sector is not differentiated between, as nodes expose themselves directly to attackers in terms of perceived utility. The source sectors of attacks is also not included, as this directly stems from the attacker types identified in sub-section 3.2.1. This results in the following classification scheme:

1. Impact/motivation
    a. Disrupt
    b. Destroy
2. Method of operation
    a. Isolated denial of service
    b. Integrated worm

Whereas Miller and Rowe (2012) identify eleven attacks methods, for this ecosystem these are grouped together into two categories. The first method is an isolated denial of service attack, which seeks to damage or disable a single infrastructure node. This method of operation does not limit itself to typical distributed denial of service attacks, but encompasses any targeted attack that seeks to cause damage. The other category is integrated worm attacks, comprising of attacks that are capable of spreading if not attended for. Based on this scheme, three types of attacks are identified, corroborated with academic literature:

1. Disruptive malware – Disrupt, Integrated worm
   Making use of common malware to infect information systems within critical infrastructures. The most prominent recent example of this type of attack was a large-scale ransomware attack on British health infrastructure, affecting dozens of hospitals (Clarke & Youngstein, 2017). While these attacks are limited in power, they are capable of spreading throughout the interconnected network if left unattended.
2. Infrastructure blackout – Disrupt, Denial of service
   The second type of attack involves denial of service-based attacks with the main purpose of disrupting infrastructure operability. The associated power of this attack is moderate, as the attack seeks to inconvenience infrastructure more than causing physical damage or harm. Yuan, Zhu, Sun, Wang, and Basar (2013) identify several of these attacks with the main trend in their increase being linked to the ease at which these can be conducted. An example of a high-end version of this attack vector was the 2015 cyberattack on a regional Ukrainian power grid, where data injection attacks managed to disable core facilities for a prolonged duration (Liang et al., 2017).
3. Infrastructure asset destruction – Destroy, Integrated worm
   The last type of attack is aimed around causing physical damage to infrastructure assets, with the intent of destruction. The most prominent example of this attack type is the Stuxnet worm that was able to spread across the world, believed to have originally targeted Iranian nuclear facilities Farwell and Rohozinski (2011); Karnouskos (2011). These attacks are considered the most powerful, but require very sophisticated and elaborate preparation.

## 3.3 Defensive strategies and control mechanisms

In order to cope with the increased frequency and impact of cyber-threats to CI systems, infrastructure operators make use of multiple control mechanisms to secure functional operation of

CIs. This section will provide insight into the formulation of coherent defensive strategies, consisting of a multitude of control mechanisms. First, the overall cycle by which control mechanisms are applied is described. This is followed by the specification of preventive mechanisms, intrusion detection mechanisms and responsive measures respectively.

### 3.3.1 Control cycle

Defensive strategies are defined as configurations for control mechanisms used to thwart and mitigate cyberattacks. Control mechanisms are individual measures that attempt to reduce the impact inflicted by cyberattacks. The allocation of control mechanisms impacts the ability for defenders to respond to a versatile *threat environment* or *threat landscape*. The FAIR framework by Jones (2006) proposes three purposes for control mechanisms:

1. *Preventive controls* seek to filter illegitimate traffic before it enters the system
2. *Detective controls* seek to detect whether any undetected threat has managed to bypass preventive controls
3. *Responsive controls* seek to prevent escalation of damage for detected intrusions

Within the context of CI systems, the same distinction is proposed by Cárdenas et al. (2011) and commonly dissected for research on specific control mechanisms (Berthier, Sanders, & Khurana, 2010; Linda, Vollmer, & Manic, 2009; Pasqualetti et al., 2013). An important additional factor is to establish an accurate assessment of expected impact (Douligeris & Mitrokotsa, 2004; Ten et al., 2010). These impact tests are used to determine the appropriate responsive measure.

A defensive strategy describes the extent to which these controls are implemented. Each type of control mechanism provides benefits, but as no security implementations are perfect, there is also a possibility of false results, leading to false negatives or false positives (Patel, Taghavi, Bakhtiyari, & Júnior, 2013). These will be related to instances of each control mechanism in the next three sub-sections. The chronological order associated with defensive strategies is as follows: prevention mechanisms are applied when inbound traffic is received by an infrastructure node. Prevention mechanisms judge whether the traffic should be blocked or not. Detection systems operate as a second line of defence against unprevented intrusions. Based on the classification, an impact assessment establishes the perceived operability of a node. This perceived operability is used to determine the correct type of response. After a threat is dealt with, the degree of perceived impact is updated. This process is visualised in Figure 3-2.

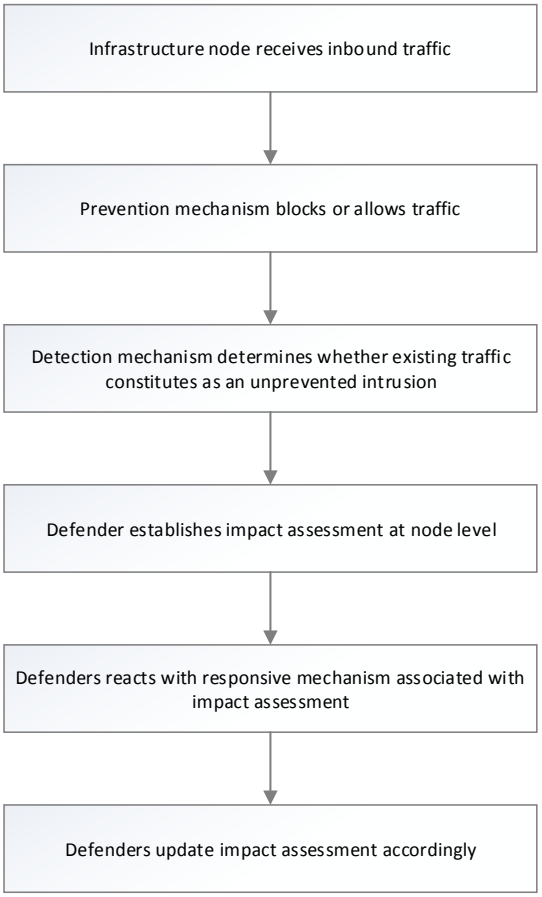


*Figure 3-2: Chronology of defensive strategies*

### 3.3.2 Prevention mechanisms

The first line of defence is formed by prevention mechanisms, which seek to discern between legitimate traffic and malicious attacks (Li et al., 2012). There are multiple means of achieving this, but the main task is always similar on an ecosystem level. While just one facet of defensive strategies, prevention mechanisms pose the most direct frontier for cybersecurity challenges. Failure to prevent attacks can pose severe consequences, leading to often stringent requirements being placed on prevention systems (Korobiichuk, Hryshchuk, Mamarev, Okhrimchuk, & Kachniarz, 2018). Prevention can be filter-based or authentication-based. The former implies classification of all traffic going into the system and excluding predetermined entities (Li et al., 2012). The latter requires all possible activities of the system to be specified upfront, as only traffic that matches the prescription is allowed (Douligeris & Mitrokotsa, 2004). With growing demands for openness of critical infrastructures, purely authentication-based systems do not possess the required agility to navigate the growing threat landscape (Douligeris & Mitrokotsa, 2004; Li et al., 2012).

Intrusion prevention systems effectively determine further choices made as a result of the classification made (Patel et al., 2013). By establishing the direct interpretation of traffic, the reaction chosen in turn directly impacts the operability of an infrastructure node. Allowing an attack through the system or erroneously blocking legitimate user traffic can both negatively affect the entire threat landscape, as consequences from attacks can lead to cascading failures in dependent nodes, as well as worm-based attacks spreading through connected nodes (Staniford-Chen et al., 1996). Requirements for accurate predictions are especially stringent because there is only one chance for prevention mechanisms to generate a classification. The aim for designing prevention mechanisms is

therefore based on sensitive positive results (correctly classifying an attack) as well as being specific in classifications (not classifying innocuous traffic as an attack) (Elhamahmy, Elmahdy, & Saroit).

### 3.3.3 Detection mechanisms

The second line of defence is comprised of intrusion detection systems. Intrusion detection systems are continuously applied as part of ongoing monitoring processes, whereas intrusion prevention is only applied when traffic is encountered (Ten et al., 2010). Essentially, detection mechanisms relate to the current state of infrastructure nodes in order to specify whether there are active attacks, and if so, the number and severity of attacks. There are several approaches to implementing intrusion detection: *signature-based*, *specification-based* or *anomaly-based* (Berthier et al., 2010; Mitchell & Chen, 2013; Ntalampiras, 2015). Anomaly-based intrusion detection seeks to establish entities operating outside of predefined behavioural rules (Mitchell & Chen, 2013). Whereas this performs better in adapting to new or unknown attacks, this also tends to yield higher false positive rates (Elhamahmy et al.; Mitchell & Chen, 2013). Conversely, signature-based and specification-based intrusion detection approaches both attempt to match entities with predefined properties of known attacks (Staniford-Chen et al., 1996). This works well at establishing what kind of attack is conducted, but often falls short at detecting new types of attacks (Ntalampiras, 2015). This can fail to detect newer types of attacks, but ensures a minimum of false positives.

Based on the assessment made by intrusion detection systems, infrastructure operators judge the perceived operability. This feeds into their decision-making which will be further elaborated upon in the next sub-section. False negatives (failing to detect an attack) then imply an underestimation of active threats, whereas false positives (incorrectly detecting a non-existent attack) result in an overestimation of perceived inoperability. For these reasons, intrusion detection systems are subject to a substantial number of scientific studies (Amin et al., 2013b; Berthier et al., 2010; Cárdenas et al., 2011; Elhamahmy et al.; Linda et al., 2009; Miciolino et al., 2017; Mitchell & Chen, 2013; Pasqualetti et al., 2013). Within the scope of this study however, it is mainly the first elements that are of interest: being able to classify attacks and harmless legitimate traffic in such a way that decision-making processes could be altered. Combining these elements with the presence of dependencies further affecting infrastructure operability results in a rich mix of ecosystem elements for a simulation model.

### 3.3.4 Impact testing and response

The last element of defensive strategies rests in the assessment of current operation and the associated responsive mechanisms. Based on the impact assessment, decisions are made with regards to expected outcomes (Ten et al., 2010). Different security scenarios call for different measures, as certain scenarios result in higher or lower susceptibility to threats. Conducting impact tests is a key element of the cybersecurity ecosystem, as specifically the deviation between perceived impact and actual impact can substantially shift system behaviour (Charitoudi & Blyth, 2014).

Two types of responses are identified, besides doing nothing: *alleviating* and *retaining* intrusions (Asnar & Giorgini, 2006). Alleviation implies keeping a node active while trying to remedy active intrusions. In case the failure was inflicted by attackers planting a worm, this could possibly lead to an infection spreading through the network (Staniford-Chen et al., 1996). Retention of intrusions means the disconnection of a node, as no countermeasure could prevent further damage at this point. This could result in cascading failures in connected nodes, but ensures that no further damage can be sustained. Similarly, failing to properly deal with ongoing intrusions could lead to the same extent of cascading failures, or even worse if attacks drag on for long enough. Impact testing and subsequent decision-making is a crucial process in the ecosystem. Establishing thresholds for certain decisions can help generate insight into exploratory system behaviour under different scenarios.

Cybersecurity is not exclusively limited to only attacks and prevention, as the pace at which responses are made can also indicate crucial elements of system performance.

## 3.4 Intermediate findings

Throughout this chapter, cybersecurity elements related to the ecosystem of critical infrastructures were laid out and discussed. The aim was to answer the second research sub-question, establishing control mechanisms and attacker properties and how these affect cyber-risk within the ecosystem. The first section laid out definitions of core cybersecurity elements such as cyber-risk itself and why the common definition of cyber-risk is not fully relevant to this study. Furthermore, it was found that several cybersecurity approaches and frameworks require slight adaptation to fit the scope of critical infrastructures. The second section detailed a derived taxonomy of cyberattackers and cyberattacks. This was done to generate insights into the effects caused by certain attacks, crucially identifying differences between worm-based attacks and denial of service-based attacks. The third section denoted elements of defensive strategies, which will be crucial for model experimentation phases. It was found that defensive strategies consist of several parameters for deployment of *prevention mechanisms*, *intrusion detection mechanisms* and *impact testing and responsive mechanisms*. These elements will primarily form the input for the eventual agent-based model. Updating the conceptual overview of ecosystem elements, the properties discussed throughout this chapter were added to the model. This is shown in Figure 3-3, with changes highlighted in blue.

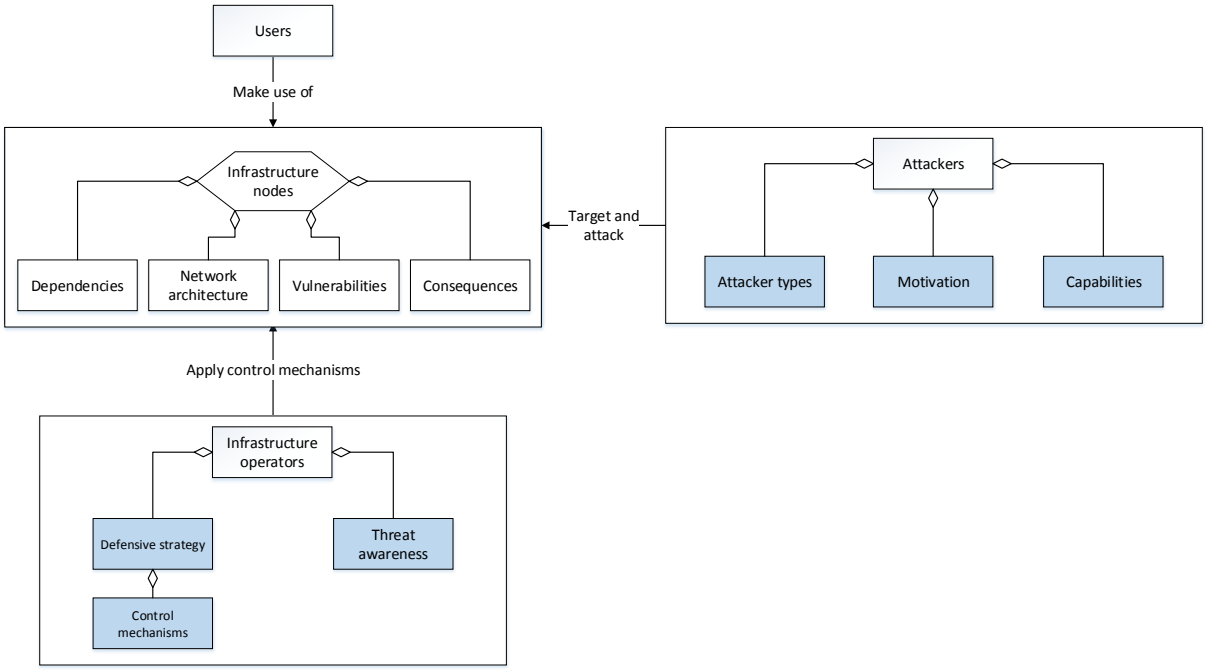


*Figure 3-3: Expanded conceptual model with new additions highlighted in blue*

# 4 Attacker and defender behaviour in the ecosystem

This chapter describes properties related to attacker and defender behaviour within the cybersecurity ecosystem for critical infrastructures. These concepts specify constraining elements and driving forces behind interaction among actors within the ecosystem. The main objective for this chapter is to formulate an answer to the third research sub-question, which was defined as follows:

*Which properties for attacker and defender behaviour aptly describe decision-making behaviour in the cybersecurity ecosystem of critical infrastructures?*

The elements discussed in this chapter relate to attacker and defender behaviour. Expanding on the conceptual overview established throughout previous chapters, Figure 4-1 depicts elements that are to be expanded upon. The first section provides insight into information as a constraining factor to actor interaction and decision-making. The second section details how infrastructure operation is disrupted and how operators assess this level of operation. The third section wraps up this chapter, establishing intermediate findings required to answer sub-question 3.

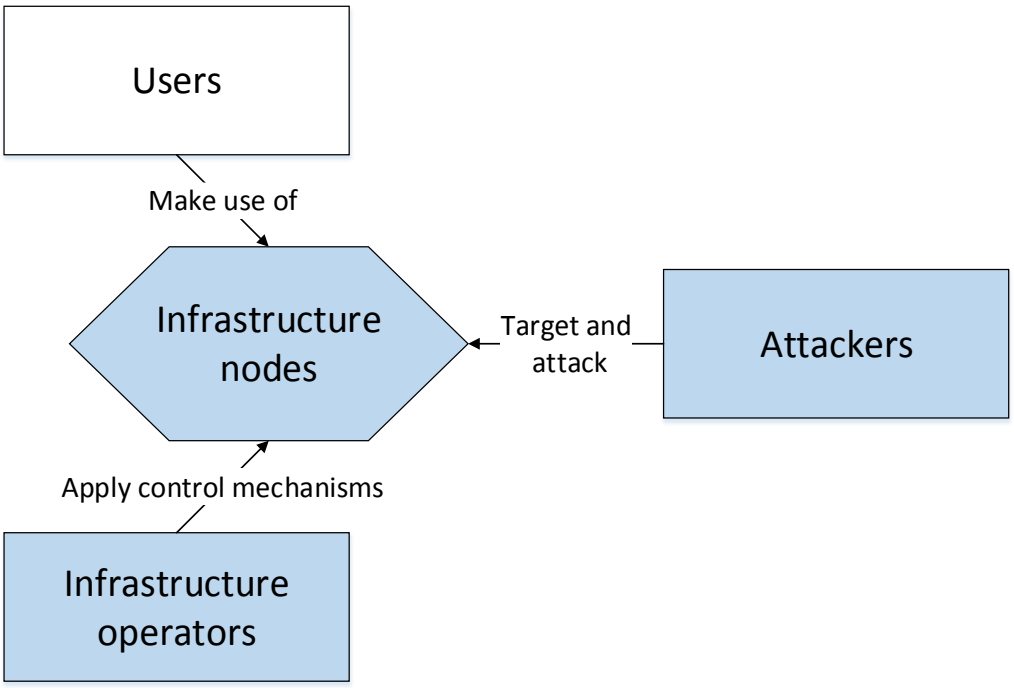


*Figure 4-1: Highlighted elements from the conceptual overview to be elaborated throughout this chapter*

## 4.1 Situational awareness

The first element discussed involves actions and interactions related to both attackers and infrastructure operators. The degree of situational awareness affects the capabilities of actors to make rational decisions (Liu et al., 2012). First, limitations to defender decision-making will be discussed. Secondly, limitations to attacker decisions will be discussed.

### 4.1.1 Defender information

Infrastructure node operators make their decisions based on an assessment of current threats to infrastructure nodes. As stated in chapter 3, impact tests directly feed into the establishment of active intrusions to the network. This awareness is used to establish appropriate responses, within the context-dependent observations of a defender (Sridhar, Hahn, & Govindarasu, 2012). Rybnicek et al. (2014) discuss the differences in attacker and defender views, with core emphasis placed on the process by which infrastructure operators decide on their responses. The process of assessing the

cause and effect of certain impact scenarios involves indexing information related to vulnerabilities that expose attack vectors to external threats as well as indexing information related to the system's state of operation (Ten et al., 2010). Typically, studies model cybersecurity scenarios through game theoretic approaches, where attackers and defenders both attempt to optimise their utility by reacting to the information presented by the other party (Brown et al., 2006; Hare & Goldstein, 2010). Brown et al. (2006) distinguish between attacker-defender models and defender-attacker-defender models for CI systems. The former implies defenders responding to threats initiated by attackers, whereas the latter enables a changing deployment of defensive resources. The conceptual model established in this study conceptualises a defensive strategy as the static configuration of control mechanisms. As such, for the purposes of this study, behaviour is characterised as an attacker-defender model, as the deployment of defensive resources is considered a static element in the ecosystem. Instead, attackers and defenders both act on information available to them to determine their best action. Further game theoretic expansion is therefore not required, since the availability of information can help operationalise interaction for static environments.

In essence, defender information is constrained to their awareness of active threats to the system. This cultivates a perception defined as *situational awareness* (Alcaraz & Lopez, 2013). The degree of situational awareness describes the encompassing view of defenders on the threat landscape. The characterisation depends on a combination of three elements, as per Alcaraz and Lopez (2013):

- *Infrastructure anomalies*: identification of physical events and whether observed activities are within permitted thresholds
- *Anomaly control:* Detecting malfunctions in (physical) node operation within the control network.
- *Intrusion control:* Detecting suspect activities within the control network

Relating to cybersecurity elements, only intrusion control is relevant for critical infrastructure operation. Infrastructure anomalies and anomaly control both refer to physical elements that are irrelevant to decisions relating to cyber-activities. On the other hand, intrusion control directly establishes the degree of cyber-situational awareness (Alcaraz & Lopez, 2013). This degree of situational awareness is used to anticipate, detect and respond to anomalies in network operation. Assessments are made within the network to identify and assess the capabilities of attackers, as well as the presence of attack-related anomalies.

The degree of situational awareness is established by defenders' capabilities of detecting intrusions and anomalies while not creating false alarms (Vasilomanolakis, Karuppayah, Muhlhauser, & Fischer, 2015). The most prominent element for establishing awareness of threats to the system is the effectiveness of intrusion detection systems, which in turn is based on continuous monitoring activities (Patel et al., 2013). Activities pertaining to intrusion detection seek to primarily detect *misuse* and *anomalies*. Regardless of the type of activity applied, the result is a decision to flag deviations. This deviation is then used to classify which type of attack is detected, which is used to establish the perceived threat to the infrastructure node (Alcaraz & Lopez, 2013; Cárdenas et al., 2011).

The impact assessment associated with attack classification is then used to compare the perceived threat to the system to certain response thresholds (Linda et al., 2009; Sridhar et al., 2012). The fuzziness of impact assessment based on situational awareness results in rational actor-level decisions, as they follow their own set of rules for their context-dependent awareness, yet irrational decisions from an encapsulating observer's point of view (Rybnicek et al., 2014). Since intrusion prevention and detection systems are not perfect, the degree of situational awareness often diverges

from the true presence of threats for a network. A core element of cyber defensive strategies is therefore to assess which configurations for intrusion detection systems yield the highest degree of accurate operational decisions (Yuan et al., 2013).

Essentially, intrusion detection and prevention systems are used to identify two types of events, regardless of whether these events correspond with misuse or anomaly identification (Patel et al., 2013). These are as follows:

1. Active attacks
    a. If intrusion detection or prevention results in an detecting an attack (positive classification), resembling a true positive
    b. If intrusion detection or prevention fails to detect an attack (a negative classification), resembling a false negative
2. Harmless, legitimate system operation
    a. If intrusion detection or prevention results in detecting an attack (a positive classification), resembling a false positive
    b. If intrusion detection or prevention results in not detecting an attack (a negative classification), resembling a true negative

Performance metrics for intrusion detection are typically associated with these concepts, described by the false positive rate, which is related to the specificity of a metric, and false negative rate, which is related to the sensitivity of a metric (Patel et al., 2013; Vasilomanolakis et al., 2015). The undesired outcomes for defenders, false positives and false negatives, directly contribute to differences between perceived threat awareness and true threat presence. However, there are slight differences between effects of intrusion detection and prevention misclassifications. For intrusion detection, false positives lead to a higher estimation of threats, whereas false negatives lead to an increase in true presence of threats. For intrusion prevention on the other hand, false positives result in blocking access for legitimate traffic, causing a loss in operability within the system and thus the true presence of threats, while not changing the context-dependent awareness. This occurs because defenders perceive a threat to have been dealt with, not realising that they blocked access from a non-threat which results in direct efficiency losses (Alcaraz & Lopez, 2013; Puig, 2018).

### 4.1.2 Attacker information

The other actors that act within the boundaries of their own context-dependent awareness are attackers. The inclusion of awareness-related concepts are slightly more subtle than with defenders and primarily pertain to target and attack selection. The central element to attacker decision-making is utility-maximisation: attackers seek to attain maximum perceived utility and will find a target to attack accordingly (Brown et al., 2006). Crucial to this notion is the limitations they experience in terms of available and accessible information (Janssen & Sharpanskykh, 2017).

Besides assessing threats and controls, attackers within this ecosystem require further specification (Byres, 2004). The main addition made is the perception of *threat attractiveness,* which serves as the primary motivation for whether an attacker decides to initiate an attack or not. While Byres (2004) specifies that such elements are difficult to operationalise for cybersecurity issues, the components of critical infrastructure losses established in chapter 2 (physical and economic impact) and attack motivations discussed in chapter 3 enable a classification scheme that incorporates attacker preference (Miller & Rowe, 2012). Attackers assess their threat attractiveness in terms of preference for physical and economic losses resulting from an attack. In the light of game theoretic models touched upon in section 4.1.1, attacker actions follow maximisation problems, where they attack a target following individual assessment of threat attractiveness.

This component of attacker information is further emboldened due to lacking knowledge of the target population, in this case infrastructure nodes. There is a certain degree of ambiguity in decision-making for critical infrastructures, where a lack of awareness of current context might lead agents to make suboptimal decisions (Charitoudi & Blyth, 2014). There is a certain level of knowledge for attackers, depending on their profile (established in chapter 3). This level of knowledge affects the level of refinement applied during target selection. Attackers lacking information on the system might apply target selection randomly, whereas more sophisticated target selection mechanisms involve optimising perceived utility following their loss preferences. Even greater knowledge of the system could allow for attackers assessing further impact due to cascading failures in outbound dependent nodes. All in all, attackers are just as limited in terms of their situational awareness as defenders (Janssen & Sharpanskykh, 2017). However, Brown et al. (2006) identify attackers as being in an advantageous position in terms of information availability, since defenders have already applied their defensive strategy and do not make any strategic decisions in response to attackers. In essence, attackers make proactive decisions to maximise their risk based on their predetermined attack configuration and defenders react to these decisions, establishing perceived threat awareness and picking responsive mechanisms accordingly.

## 4.2 Infrastructure operability

Another constraining factor to interaction among the system is the core concept of infrastructure node operation, or operability. This concept was mentioned in prior sections of this study, but requires further elaboration now that other factors have been established. First, the main concepts pertaining to infrastructure operability are expanded upon. These are subsequently linked to infrastructure states and their relation with dependencies is discussed.

### 4.2.1 Definition of operability

The main definition of operability, the mode of operation, rests in the efficiency attained to within an infrastructural node (Puig, 2018; Setola & Theocharidou, 2016). The degree of operability serves as a universal metric for infrastructure node behaviour, and can for this reason be used to define the effects of dependencies contributing to cascading failures (Rinaldi, 2004; Setola & Theocharidou, 2016). Since the granularity of analysis for this study is aimed around assessing ecosystem level effects and mitigation of undesired effects through defensive strategies, applying this universal definition opens the door for defining a concept applicable to most interaction.

The notion of operability relates to the sensitivity of system operation, and by extent affects the degree to which losses are incurred (Puig, 2018; Sridhar et al., 2012). Infrastructure node loss consequences can be described as a degree of losses incurred respective to the degree of inoperability, the extent to which operation is hampered (Rinaldi et al., 2001). By incorporating the operability of an infrastructure node as a common factor, this can be used to enable actor interaction around a central concept (Rinaldi, 2004). Successful cyberattacks and erroneously blocked user traffic tarnish the mode of operation within an infrastructure node, which in turn affects the mode of operation in dependent nodes. On the other hand, the impact assessment of defenders can also be related to the level of operability of an infrastructure node. They assess the expected impact of the perceived attack to establish a level of perceived operation, which in turn is related to thresholds for responses.

### 4.2.2 System operational states

While the operability of a node should be included as a continuous scale, as per Puig (2018), the interpretation of values along this scale should depend on several definitions. Setola and Theocharidou (2016) define four states for system operation, which are shown in Table 4-1 below.

*Table 4-1: Infrastructure node operability states and their definitions, adapted from Setola & Theocharidou (2016)*

| *State* | *Description* |
|---|---|
| *Normal* | The state in which a critical infrastructure node operates under normal operational conditions. If there is any threat, it is very minor in terms of impact. |
| *Stressed* | The state in which a critical infrastructure node operates when special measures should be taken to keep system operation in control. Threats can be minor and major inconveniences, but do not single-handedly disable the node. |
| *Crisis* | The state in which a critical infrastructure is destabilised and out of control. Significant losses are sustained and operators should detect this and react appropriately. |
| *Recovery* | The state in which a critical infrastructure is closed off from the network in order to reconstitute system operation. This is achieved by retaining intrusions, as discussed in chapter 3. |

These states represent the meaning of system operability and define the type of response typically associated with a certain level of operation. While there are other states possible depending on the refinement of specific infrastructures, this selection of states can be used to visually represent activities associated with nodes in certain conditions. Besides, it can highlight how differences between perceived operability and true operability emerge as the result from detected and undetected attacks.

The overall impact of infrastructure operability is a central element for actor interaction across the ecosystem. Operability is used to assess the extent to which losses are incurred, as well as identifying defenders' impact assessment. The mode of operation, as such, helps operationalise several elements that were originally considered problematic for quantification.

## 4.3 Intermediate findings

Throughout this chapter, elements related to actor interaction were discussed. Specifically, these interactions related to actions taken by attackers and defenders. The aim for this chapter was to formulate an answer to the third research sub-question, assessing which properties describe attacker and defender behaviour. The first section identified crucial elements in defender and attacker decisions, indicating how both operate on a degree of *situational awareness*. It was found that both entities are constrained due to the limited availability of information, with attackers carrying an advantage over defenders, since they get to optimise utility gains based on fixed loss parameters for infrastructure nodes. Defenders are constrained due to limitations in impact assessment, which can lead to counterproductive responses. In short, attacker and defender behaviour is inherently context-dependent, whereas system configurations such as defensive strategies are static. As such, game theoretic models would not add value within the scope of this study. The second section identified how the notion of *infrastructure operability* can be used to formalise and operationalise the effects of cyberattacks and dependencies among an ecosystem of infrastructures. A classification scheme is provided that distinguishes node operation states as *normal, stressed, crisis* or *recovering*. These elements are applied to existing concepts within the ecosystem to be formalised within the model. The added elements are shown in Figure 4-2, with changes highlighted in blue. This concludes the system specification phase of this study, leading into the model conceptualisation and formalisation phase.

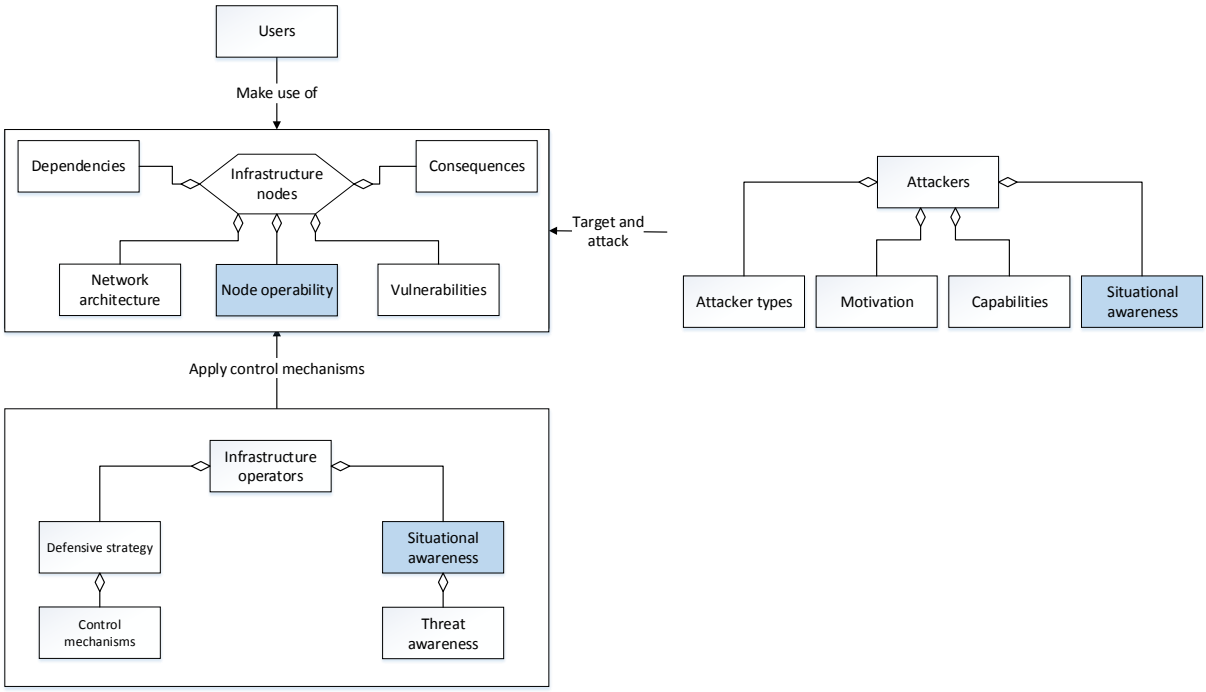


*Figure 4-2: Overview of elements added to the conceptual overview throughout this chapter, resulting in the finalised ecosystem-level aggregation model*

# 5 Conceptualising an ecosystem model of critical infrastructures

This chapter seeks to establish a conceptual model for the cybersecurity ecosystem of critical infrastructures. This step in model development is associated with the *System Identification and Decomposition* and *Concept Formalisation* steps discussed by Nikolic et al. (2013). This phase is crucial for model development, as all necessary elements are incorporated and detailed to prepare for model implementation. Specifying a conceptual model involves integrating all elements discussed in chapter 2, chapter 3 and chapter 4 following an integrated framework. The framework applied in synthesising the identified concepts into a coherent model is complex adaptive systems (CAS) thinking, which will be discussed in section 5.1. The perspective of CAS thinking on the cybersecurity ecosystem of critical infrastructures is provided in section 5.2, providing insight into how a conceptual model can be used to answer research questions. Section 5.3 further specifies the ecosystem model, linking elements identified in previous chapters to CAS concepts. Section 5.4 provides a set of performance metrics that can be used to assess different elements of system performance. The chapter is wrapped up and summarised in section 5.5, providing key pointers for model formalisation in chapter 6.

## 5.1 Complex adaptive systems: a definition

The essence of the approach taken for this study is nested in complex adaptive systems thinking. First, the background of CAS will be described. Subsequently, an integrated definition will be provided. Afterwards, the main properties of CAS are discussed.

### 5.1.1 The background of complex adaptive systems

Nikolic and Kasmire (2013) propose a framework for understanding CAS and translating this understanding into agent-based models. CAS thinking is in itself an adaptation from traditional systems thinking, which is identified by Ryan (2008), citing W. Ross Ashby, as being "*a set of variables sufficiently isolated to stay discussable while we discuss it*". This definition of systems forms the foundation of CAS thinking. Crucially, Ryan (2008) identifies, among others, *organisation, interdependent components, interaction with their environment* and *emergence* as key properties of system components.

Complex adaptive systems, as the name suggests, extend beyond the definitions adhered to within traditional systems thinking. CAS differs as the systems are perceived to be both *complex* and *adaptive*. Adaptiveness in the context of CAS relates to improving system behaviour over time, whether through physical, social, technical or cultural shifts in the environment (Nikolic & Kasmire, 2013). A system that is adaptive does not merely change over time, but instead shows improvement due to a change in context awareness or learning behaviour. Because all actions within a complex adaptive system are based on the context of a given situation, it is almost impossible to predict the state the system has evolved towards. The essence of adaptiveness is nested in a large number of actors constantly interacting, each changing their state accordingly.

The other notion, *complexity*, is more sensitive to applicable and contemporary worldviews. Complexity can be described as the property of not being simple, but often requires further specification of complicated and complex aspects (Nikolic & Kasmire, 2013). Zadeh (1973) relates complexity to the ability to describe system behaviour under different circumstances. Being able to accurately predict system behaviour implies a foundation of understanding of driving forces system-level behaviour. Complex systems can be understood, and therefore modelled, provided that a fitting formalism is specified (Mikulecky, 2001). A system is considered complicated if this degree of understanding is not present: for a given scenario *X* it is almost impossible to correctly predict consequence *Z* associated with action *Y*. Managing complexity in systems requires the application of

multiple formalisms to understand which properties affect system behaviour and to carefully distinguish between relevant and irrelevant elements (Nikolic & Kasmire, 2013).

### 5.1.2 The complex adaptive systems paradigm

The CAS paradigm is a branch of system thinking that integrates the three main concepts discussed previously: it requires researchers to identify formalisms to cope with complexity, identify the adaptiveness of system behaviour as well as identifying components and interactions among the system of interest. John H. Holland in Waldrop (1992) defines complex adaptive systems as: *"a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing. The control of a complex adaptive systems tends to be highly dispersed and decentralised. If there is to be any coherent behaviour in the system, it has to arise from competition and cooperation among the agents themselves. The overall behaviour of the system is the result of a huge number of decisions made every moment by many individual agents."*

Crucially, CAS are defined by bottom-up interaction that shapes system-level behaviour. The only facets that need to be implemented relate to individual entities making decisions. The way these entities interact is therefore critical to understanding system behaviour. This perspective on systems evolved significantly from the systems thinking paradigm as described by Ryan (2008), highlighting the essence of system-level behaviour is in itself inherently rooted in the parts that compose a system.

### 5.1.3 Key properties in complex adaptive systems thinking

CAS are characterised by the presence of chaos in agent interaction and emergent behavioural patterns as a form of overall system behaviour (Nikolic & Kasmire, 2013). Chaos implies a degree of changeability depending on a set of conditions at any point in time. This relates back to the notion of complexity provided by Zadeh (1973). Chaotic systems do not necessarily imply randomness, as subtle differences in starting configurations or encountered scenarios can make agents shift their behaviour. The second key concept of CAS, emergence, stems directly from decentralised agents making decisions based on their contemporary environment. Emergence is the consequence of bottom-up decision-making behaviour leading to coherent system behaviour (Nikolic & Kasmire, 2013). Crucially, emergent patterns are directly tied to the combined presence of all parts in a system. Emergent patterns cannot be explained by isolated, individual agent rules and directly stem from dynamic interaction of the network of many agents as defined by Holland (Morin, 1999). An aspect of emergent patterns is the capability of CAS to self-organise. The similarity of these properties to those presented by Ryan (2008) highlight how the CAS paradigm is an evolution of traditional systems thinking.

## 5.2 Modelling the critical infrastructures ecosystem

The ecosystem of cybersecurity for critical infrastructure systems, as discussed throughout this study, can be classified as a complex adaptive system for a multitude of reasons. For this classification to be eligible, a number of conditions described in section 5.1 should be met. After discussing these conditions, the main method is argued for.

### 5.2.1 Critical infrastructures as a complex adaptive system

First, it should be established that cybersecurity for CI systems largely corresponds with systems thinking. As discussed in chapter 2, critical infrastructures are vast networks of heterogeneous infrastructure nodes. These system artefacts consist of a wide variety of different elements, with specific cybersecurity decisions being made at the node-level. Decisions related to defensive strategies tend to be made at higher institutional levels, meaning that interaction is governed on

multiple levels (Van Dam, Nikolic, & Lukszo, 2012). Cyberattackers and infrastructure node operators each make decisions directly tied into cybersecurity elements: cyberattackers make decisions related to utility optimisation and information gathering as discussed in chapter 4 and operators make decisions related to infrastructure operation and traffic classification. Whether attacks are thwarted or successful relies on decisions made by a multitude of agents, as well as environmental context. While the process of attacking and defending is generally well-known, the intricate balance that takes place between these actions is academically unexplored on an ecosystem-level. This level of analysis, or worldview in the words of Nikolic and Kasmire (2013), assesses dynamic interaction that extends beyond the objectives for individual node operators, who seek to only assess their optimal defensive strategy.

The second qualification to be met is adaptiveness. Critical infrastructures themselves have shown to adapt significantly over the past decades, as discussed in chapter 2. However, this adaption primarily originates from the need to cater towards a changing environment. The true nature of adaptiveness takes place in terms of impact assessment with regards to interdependent node operability. Agents within the ecosystem are constantly seeking to achieve improvements for their own goals, whether malicious or not. Cyberattacks have been getting more and more sophisticated, and infrastructure operators are constantly adapting their situational awareness in order to adapt to a new situation. This perception of impact and operability feeds into their decision-making process, as more dangerous situations ask for more rigorous defensive decisions.

The third qualification revolves around the complexity of the system. While existing studies into control effectiveness for CI systems establish their effectiveness in a closed environment, the overall cybersecurity picture involves multiple factors beyond an individual node, as discussed in chapters 2 and 3. Individual decisions and the behavioural rules that lead to them can be identified, showcasing the processes that take place within this complex environment. However, assessment of an ecosystem-level cybersecurity environment involves inclusion of the multiple different cybersecurity facets discussed in chapter 3 and elements to interaction detailed in chapter 4.

As a result of the cybersecurity ecosystem for CI systems corresponding with the three qualifications for CAS, the resulting interaction is also characterised by CAS properties. Chaos, emergent behavioural patterns and self-organisation can all be identified within the ecosystem. Cyberattacks against CI systems are not conceived by pure randomness: there is always an instigating factor. Whether the stars align for an attacker to be able to successfully launch an attack depends on multiple factors: whether they have the resources and knowledge available to launch an attack, whether there is a viable target and whether attacks can bypass several checks in the form of defenders' control mechanisms. Chaotic behaviour occurs based on the wide variety of possible events that lead to one event or another. These decisions all emerge bottom-up, based on behavioural rules set by each agent. The likelihood of successful attacks taking place increases if all defenders fail to accurately assess situational inoperability. Additionally, lacklustre defensive decisions can quickly lead to cascading failures throughout a vastly interdependent and interconnected network of infrastructure nodes. Disjoined, local decisions lead to actions from other agents that combined lead to coherent system behaviour.

### 5.2.2 Agent-based modelling concept

This section details the chosen approach to create a simulation model for the ecosystem discussed in section 5.2.1. The first sub-section details the basic elements of an agent-based model, following the same framework by Nikolic and Kasmire (2013) used throughout this chapter. The second sub-section describes why ABM will be used, followed by the main objectives for the model in the third sub-section.

*The essence of agent-based modelling*

While there are multiple methods to model CAS, this study will make use of agent-based modelling (ABM). Agent-based modelling is considered an extension of CAS, as it allows for the model itself to be a CAS in and of itself (Nikolic & Kasmire, 2013). ABM is based on generating answers to the core question "*What happens when …?"* in an attempt to generate insight into the effectiveness of the system in certain scenarios. These questions are exploratory in nature and seek to assess the response of certain system elements to changes in environmental drivers. This underlines the objectives of exploring system behaviour given different scenario configurations. As Nikolic and Kasmire (2013) describe, there is no desired system state to achieve, as ABM revolves around finding out what happens if the system is exposed to given system configurations. On the other hand, the type of results ABM yields are by no means direct quantitative, reliable predictions for exact system performance. Instead, ABM performs well at discerning emergent patterns and behavioural tendencies from simulation and experimentation iterations.

Key elements of complex adaptive systems as detailed in section 5.1 are the local-level decisions made by individual agents, who form the central element of ABM. They make their own decisions based on predetermined rules and adaptive agent states. Nikolic and Kasmire (2013) identify these elements as agent states, behavioural rules and interactions. *Agents* are entities present within the demarcated system boundaries, who operate in a decentralised, independent and context-sensitive manner.

Besides agents, ABM also operates within a specified model environment, which encompasses all elements in the model, as well as external variables that entities within the model could interact with. The model environment embodies the physical and/or logical location of agents, as well as all relevant external elements required for their interaction. By extent, the model environment includes all agents and all elements possibly necessary to facilitate interaction within the chosen formalisms. Another element essential to identify before devising an agent-based model is the time frame associated with the system of interest. Agent-based models incorporate a discrete time scale, whereas real-world CAS follow a continuous time scale. Since an agent-based model is expected to be a CAS representation of the original system, the limitations caused by the discrete time should be kept to a minimum (Nikolic et al., 2013). This can be achieved by modelling actions in the order and frequency by which they appear in the real-world, normalised to fit the time scale applied in the model.

*Agent-based modelling for cybersecurity purposes*

Agent-based modelling cybersecurity ecosystems has seen an uptick in recent years, as it enables modelling a great variety of cybersecurity scenarios and can form an extension of game theoretic methods (Charitoudi & Blyth, 2014; Hare & Goldstein, 2010; Janssen & Sharpanskykh, 2017; Priest et al., 2015; Rybnicek et al., 2014). Cybersecurity in itself is an emerging discipline, which leads to academics naturally asking a lot of exploratory questions along the line of "*What happens when …?"* The flexibility offered by ABM to include many facets of ecosystems and the relative simplicity of relationships and interactions helps in reducing the immense complexity of ecosystem-level problems.

Charitoudi and Blyth (2014) managed to establish a model that simulated the impact of cascading failures within an interconnected, heterogeneous network. The authors found that agent-based models work well at observing interactions and dependencies throughout critical infrastructures. Priest et al. (2015) modelled defender interactions as the redistribution of resources as a means of hindering cyberattacks' penetration, assessing the effectiveness of moving target mechanisms on a network level. The notion of redistributing resources as a defensive measure is similar to defensive

mechanisms modelled by Hare and Goldstein (2010). Janssen and Sharpanskykh (2017) took a different approach, modelling security checkpoint intrusion. The authors established means for vulnerability modelling in specific network nodes, forming a coherent overview of impact assessment in an agent-based modelling environment. Similarly, Rybnicek et al. (2014) modelled a network of individual infrastructures with distinct dependencies, highlighting the interconnected impact of cyberattacks.

Most related, non-ABM academic studies work under the assumption of rationality due to computational limitations arising from a top-down view of the system, which is circumvented within the formalisms of CAS thinking (Janssen & Sharpanskykh, 2017). Instead, emergent behaviour is used as the main proponent of susceptibility to cyberattacks following individual agent states for both attackers and defenders. As discussed in chapter 4, agent behaviour can simply not be modelled as rational due to real-world complications arising from situational awareness. Applying ABM helps reduce complexity while still being able to explore emergent patterns given different system configurations.

#### *Objectives for the agent-based model*

Following the identification and specification of crucial ecosystem elements in chapter 2, chapter 3 and chapter 4 as well as the specification of the ABM approach in this chapter, the foundation of the agent-based model can be conceptualised. In order to simulate the effectiveness of defensive strategies, an agent-based model will be created that incorporates all crucial ecosystem elements. The agent-based model will resemble an ecosystem of interdependent and interconnected critical infrastructure nodes, as well as cyberattackers and users of these nodes. The model should seek to establish a concise yet representative overview of infrastructure nodes and primarily relate to how losses are incurred and transferred among infrastructure nodes.

The main input to serve for the core question "*What happens when …?"* relates to system configurations for defensive strategies. In order to explore the robustness of defensive strategies, model parameters should be explored for possible sensitivity to attacker configurations or weightings assigned to dependencies. If these concerns are taken into account, the results from robust experimentation can be used to help establish a coherent answer to the main research question (Bankes, 1993).

## 5.3 Ecosystem conceptualisation

The first step is to identify all entities and interactions that should be included in the model. This ecosystem-level model should follow the formalisms associated with complex adaptive systems and agent-based modelling. An overview of ecosystem-level interaction between the different types of agents is shown in Figure 5-1 below. The ecosystem-level interaction model forms an extension to the aggregation model shown in Figure 4-2. The original conceptual visualisation built upon throughout this study has been refined to better match all agent-level interactions. The original model depicted in Figure 1-1 does not show the level of detail required to create a complete simulation model. The updated model forms the foundation for specification of model concepts following the specified framework.

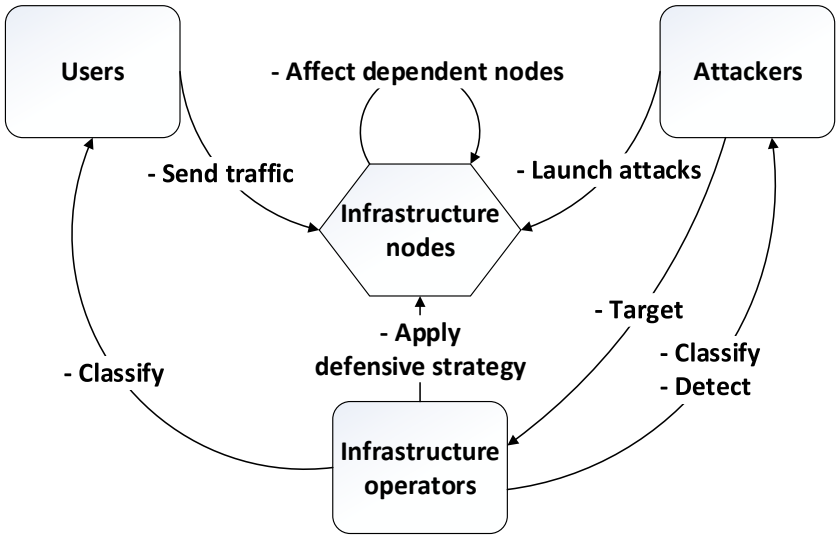


*Figure 5-1: Ecosystem interaction model*

The conceptual model depicts several core elements discussed in chapter 2, chapter 3 and chapter 4. The key entities included are users, attackers, infrastructure operators and infrastructure. The central entities around which all interaction revolves are infrastructure nodes. These nodes are subject to attacks from attackers and legitimate traffic originating from users, while affecting operability of other dependent infrastructure nodes. Infrastructure operators are responsible for protecting nodes by applying control mechanisms following the defensive strategy configuration. This defensive strategy consists of control mechanisms that seek to protect nodes by preventing and detecting intrusions and responding to current threats. Attackers target infrastructures based on their preferences and launch attacks towards infrastructure nodes.

The next step is to identify all entities that should be included in the model and determine what states, actions and rules encapsulate their behaviour. This was briefly touched upon in section 5.2 and will be expanded upon thoroughly throughout this section. Entities are placed in three categories: *agents, link entities and objects*. Typically, a fourth category for environmental factors is also included, but these are not used for this ecosystem-level study. Each entity in the model can operate based on a set of states, rules and actions, which will be further discussed throughout this section. The conceptual model has been devised in such a way that all agents feature states, rules and actions, yet objects and links are only assigned states. This is done to facilitate interaction more easily, as agents in the model are supposed to replicate their real-world counterparts in terms of independent and autonomous decision-making. An overview of model entities is shown in Table 5-1, and will be expanded upon next. The first sub-section details every agent, indicating the states, rules and actions associated with agents in this model. The second sub-section discusses the different link entities and their real-world representations. The third sub-section discusses objects and how these are to be modelled. Assumptions that were made over the course of model conceptualisation are listed in Appendix D.

### 5.3.1 Specification of agents

The first type of entity to be discussed are agents. Agents are the most refined entities within the model concept, as they are involved in context-aware decision-making processes. Each agent is tasked with a set of rules and states, as well as actions that correspond with meeting certain rules. Rules apply to multiple conditions, including environmental input, input from other agents or activation thresholds associated with an internal state. Three such agents are identified in Table 5-1.

These are expanded upon throughout this sub-section. Each type of agent will be discussed in terms of states, actions and interactions identified.

*Table 5-1: Overview of model entities, including agents, link entities and objects*

| *Model inventory* | *Description* |
|---|---|
| *Agents* | |
| *Infrastructure node operators (defenders)* | Agents who are tasked with maintaining secure infrastructure node operability. |
| *Cyberattackers* | Agents who seek to inflict damage to infrastructure nodes or the environment. |
| *Users* | Agents who make use of infrastructure nodes. |
| *Link entities* | |
| *Connections* | Physical or logical links between two infrastructure nodes that worm attacks can spread through. |
| *Dependencies* | Functional or technical dependencies between two infrastructure nodes that impede operability. |
| *Objects* | |
| *Infrastructure nodes* | Entities that together form a critical infrastructure network. |
| *Defensive strategies* | Collections of control mechanism configurations and thresholds. |
| *Control mechanisms* | Processes entities are exposed to that seek to deter cyberattacks while not impeding users. |
| *Cyberattacks* | Processes initiated by cyberattackers that inflict harm to infrastructure nodes. |

*Infrastructure node operators*

Infrastructure node operators, also referred to as *defenders*, are agents involved in operating (parts of) critical infrastructure systems. Within the cybersecurity ecosystem, they are tasked with securing key infrastructure nodes against possible cyber-threats while reducing the impact on other parties. Their task is complicated by widespread dependencies among infrastructure networks, stark consequences from minor security infractions and a limited degree of situational awareness (Alcaraz & Lopez, 2013; Lee et al., 2016; Priest et al., 2015; Teixeira et al., 2010). Their main states and actions are displayed in Table 5-2. Each of these elements has been discussed previously to some extent. States and actions related to inoperability perceptions were discussed in chapter 2, chapter 3 and chapter 4. Similarly, elements relating to defensive strategies and control mechanisms were defined in chapter 3.

As displayed in Table 5-2, defenders make use of four internal states: *perceived internal inoperability, perceived external inoperability, perceived operation* and the *deployed defensive strategy*. These four states are only the internal elements used for their actions. Besides internal states, external input also impacts decision-making. This will be discussed in the light of possible actions later. As identified in chapter 4, defenders operate based on their established situational awareness (Alcaraz & Lopez, 2013). This is represented in the conceptual model through their perception of internal and external operability. The perception of operability constitutes as their method of impact assessment, which is used to determine whether response mechanisms should be applied. The degree of perceived internal inoperability relates to awareness of threats to the internal system of an infrastructure node, for example the type of attack an infrastructure is currently experiencing. External inoperability follows from hampered operation from disruptions in other nodes, passed along through

dependencies. The deployed defensive strategy relates to the fixed values used for control mechanisms within the node. In addition, the chosen defensive strategy determines thresholds for defensive decisions. These states facilitate abstract interaction, yet together encompass the main facets relevant to decision-making. An assumption is made to use operability as the main input for decision-making, following game theoretic approaches discussed and supported in chapter 4. The main implication for this decision is that interaction does not include any strategic behaviour, while on the other hand including such concepts constrains generalisability of the model.

*Table 5-2: States and actions associated with defender agents*

| *Defender elements* | *Description* |
|---|---|
| *States* | |
| *Perceived internal inoperability* | The perceived extent of current impact on a node originating from internal inoperability, e.g. due to cyberattacks. |
| *Perceived external inoperability* | The perceived extent of current impact on a node originating from external inoperability, e.g. due to dependencies. |
| *Perceived operation* | Based on the perceived degree of internal and external inoperability, defenders make an assertion of the level of operation. |
| *Deployed defensive strategy* | The level of refinement at which prevention, detection and responsive mechanisms are deployed. |
| *Actions* | |
| *Assess current operability* | Establishing the degree of perceived operability currently experienced in a node. |
| *Classify inbound traffic* | Applying prevention mechanisms to inbound traffic, aiming to block cyberattacks and allowing user traffic. This is the event directly triggered by users or attackers interacting with a node. |
| *Block traffic* | Preventing cyberattacks access to the system, avoiding any possible damage. This is the direct result from the classification generated by *Classify inbound traffic.* |
| *Detect intrusions* | Detecting existing, unprevented attacks currently taking place. Detected intrusions will remain detected until they either expire or are removed. False positives are corrected the next time step, when intrusion detection is applied once more. |
| *Alleviate intrusions* | Attempting to remove existing, unprevented attacks while keeping the node in operation. |
| *Retain intrusions* | Removing existing, unprevented attacks while closing the node down, preventing further spread of attacks. |

As shown in Table 5-2, six actions are specified for defender agents. These actions are, as with previously discussed states, based on attack-defence properties detailed in chapter 3 and decision-making properties introduced in chapter 4. First of all, defenders create a situational assessment of perceived operability. As stated previously, this is based on an internal and an external component of operability. This assessment might deviate from the true degree of operability experienced for a defender, which could lead to counterproductive choices. The other actions are all related to the core task for defenders to distinguish between legitimate and illegitimate traffic using a prevention mechanism. This action is denoted by *Classify inbound traffic.* Inbound traffic originating from attackers and users is classified based on the prevention mechanisms applied. Accurate impact assessment and mitigation rely on accurately classifying user traffic as innocent and attacker traffic

as malicious. Traffic classified as malicious is then blocked from accessing the system. The action *Block traffic* prevents an attack from ever causing damage, but could also prevent legitimate traffic from keeping the system operational. As not all intrusions are deterred by prevention mechanisms, unprevented intrusions can be detected through the *detect intrusions* action. Based on the sensitivity and specificity of intrusion detection mechanisms applied, attacks can be detected, or false positives could generate false alarms out of nothing (Zhang, Wang, Sun, Green, & Alam, 2011). The last two actions relate to responsive measures, which are applied based on the perceived level of operation. Alleviating intrusions means removing existing attacks while the system remains operational and accessible for other nodes and traffic. Retaining intrusions also involves removing existing attacks, but closes down the node to prevent further spreading of intrusions. In both cases, the degree of operability of a node is reduced, as overall functioning of nodes is limited.

The main interactions that involve infrastructure node operators relate to dealing directly with users and attackers, as well as operating infrastructure nodes. Infrastructure nodes are the centrepiece of the model, and defenders bear responsibility for their unhindered operation. As mentioned previously, defenders are tasked with discerning between user traffic and cyberattacks, directly impacting the operability of a node. Should a node depend on other nodes, defenders need to correctly establish the possible impact that external failures might lead to. The states and actions specified to represent interaction enable operationalisation of high-level interaction, while enabling straightforward concept definitions for interaction with other agent types. Since infrastructure node operators are responsible for interaction that takes place centrally in the ecosystem model, it is important to keep conceptual definitions accurate without specifying too much information.

*Attackers*

Cyberattackers, or attackers, are agents who seek to inflict damage to critical infrastructures. As there is little interest for monetary gain, attackers seek to primarily maximise the amount of damage they can inflict. This was discussed in more detail in chapter 3. Attackers make use of advanced cyberattacks to disrupt, destroy or disable parts of critical infrastructures. Their main states and actions are laid out in Table 5-3. These states and agents were all discussed in chapter 3, except for *Knowledge*, which was introduced as situational awareness in chapter 4.

Attackers make use of four internal states following elements of the framework established in chapter 3 and chapter 4. Their *Economic loss preference* and *Physical loss preference* indicate the degree of personal utility they achieve from inflicting a certain degree of economic and physical damage, respectively. Their aims to maximise this utility is based on both economic and physical components. Attacker *Knowledge* affects the sophistication of target selection. Less knowledgeable attackers are more likely to make irrational choices, while more knowledgeable attackers can more accurately predict the possible damage caused to a node itself as well as subsequent damage caused to dependent nodes. To this end, their *Attack capabilities* delineate which attacks are available to an attacker. A knowledgeable, rational attacker with high attack capabilities might pick more powerful solutions compared to their less resourceful counterparts.

The actions applied by attackers are threefold. The first, *Find the best target,* involves the previously mentioned process of target selection. Attackers with little knowledge might select targets irrationally or chaotically, as their perception of utility could be established in a flawed manner. More knowledgeable attackers weight the expected utility from breaching a target with their loss type preferences, and in some cases assess the extension of damage through outgoing dependencies from target nodes (Byres, 2004). Similarly, the second action, *Pick the best attack*, selects the expected attack that coincides with an attacker's knowledge and attack capabilities. The background of these two actions is rooted in maximising attacker utility through maximising their preferred type of

damage. The last action, *Initiate attack*, involves the process of launching an attack on the target node. This initiates a chain of events for defenders, who have to respond to the newly arrived traffic.

*Table 5-3: States and actions associated with attacker agents*

| *Attacker elements* | *Description* |
|---|---|
| *States* | |
| *Economic loss preference* | The preference of an attacker for economic loss as the result of node inoperability. |
| *Physical loss preference* | The preference of an attacker for physical loss as the result of node inoperability. |
| *Knowledge* | The degree of refinement with which an attacker selects a target and an appropriate attack to harm their target. |
| *Attack capabilities* | The types of attacks available for an attacker. |
| *Actions* | |
| *Find the best target* | Selecting the target that is most likely to yield desired utility (loss for nodes). This is the result of their weighted expected utility for both economic and physical loss. |
| *Pick the best attack* | Selecting the attack that is most likely to inflict significant damage against the target. Attackers use more sophisticated considerations based on their level of knowledge. |
| *Initiate attack* | An attack is initiated after completion of the two prior actions. The selected target is to be attacked with the type of attack selected. This action encompasses initiating an attack, causing the defender to react. |

Attackers interact with multiple elements of the ecosystem, although they predominantly interact directly with infrastructure nodes. Attackers do not communicate or coordinate among each other, as they plan out their attack based on both their own internal states as well as the internal states of possible target nodes. Defenders do not directly interact with attackers either, as their scope of control is limited to only dealing with attacks. Within this ecosystem, defenders are not able to single out attacker agents as responsible for certain attacks, as this type of behaviour exceeds the boundaries of a cybersecurity ecosystem. While on one hand this prevents behaviour that might be observed in the real world, there was no consensus about the dynamicity of such elements of the threat landscape. For that reason, the ecosystem-level framework specifies no such concept, although this could be included as an extension of the framework for specific, practical application.

Furthermore, attackers possess the capabilities to conduct different types of attacks. In order to translate elements of the framework into complex adaptive systems constructs, attack capabilities are assigned to each attacker type. Because of the probabilistic approach, this made more sense than adding a more assumptious numerical denominator for attack capabilities. As such, there is still variety in the threat landscape for defenders to act upon based on the attack types identified in chapter 3. The types of attack associated with each type of actor are denoted in Table 5-4.

*Users*

*Users* are agents that simply make use of critical infrastructures. They represent traffic that would typically be encountered in day-to-day infrastructure operation. In this sense, users are entities that are responsible for professional usage and communication of parts of infrastructures. The role of users is more auxiliary when compared to attackers and defenders, as they mainly provide inputs for other agents to act on. The states and actions associated with users are shown in Table 5-5.

Depending on which infrastructural sectors the framework is applied to, the impact of users on operability could range from meaningless to crucial.

*Table 5-4: Cyberattack capabilities for each type of attacker*

| *Attack*<br>*Attacker* | *Disruptive malware* | *Infrastructure blackout* | *Infrastructure asset destruction* |
|---|---|---|---|
| *Cybercriminal* | X | X | |
| *Cyberterrorist* | | X | X |
| *Foreign adversary* | X | X | X |

*Table 5-5: States and actions associated with user agents*

| *User elements* | *Description* |
|---|---|
| *States* | |
| *Criticality* | The criticality of a user's traffic to a node's operability |
| *Associated node* | The specific node a user interacts with. |
| *Actions* | |
| *Send traffic* | Sending traffic to an infrastructure node as a means of regular usage of infrastructures. |

The simplicity behind user agent interaction presents itself in the scarcity of states and actions, as displayed in Table 5-5. Only two states are used by user agents: their *Associated node* and the *Criticality* of their traffic. Their associated node represents the node that functionally depends on the unhampered arrival of a specific user's traffic. A real-world example for this would be reporting by smart electricity equipment to a central load-balancing facility (Department of Homeland Security, 2015). The criticality of this traffic then determines just how crucial the presence of this traffic is to node operation. Some infrastructures might depend more heavily on a steady flow of critical information whereas other infrastructures could build a backlog of information.

The only action taken by user agents highlights their auxiliary role in the ecosystem: *Send traffic.* Given their associated node and the criticality associated with their traffic, they will transmit information related to themselves to a node. This then prompts the same response from defenders that cyberattackers instigate: classification of inbound traffic. If user traffic were to be erroneously blocked by prevention mechanisms, crucial node operation would be harmed, as discussed in chapter 3. As a result, the node's level of operation would decrease based on the aforementioned criticality of traffic. This single action also marks the full selection of interaction involving users, as they fulfil no other purpose for ecosystem operation.

### 5.3.2 Specification of link entities

The second type of entities to be detailed are link entities. These links exist as a form of connecting multiple agents for a specific purpose. Two such link entities exist within the conceptual model: *connections* and *dependencies*. Links exist to carry over information from one entity to another and serve as logical and physical input for agents' decision-making processes. Their purpose within the model will be briefly discussed throughout this sub-section.

#### *Connections*

*Connections* represent direct linkage of two infrastructure nodes. These links involve connection of two individual nodes that together shape the complex networks of critical infrastructure systems.

Relating back to the visualisation shown in Figure 2-2, connections are shown as the essence of critical infrastructure networks. These are necessary to operate infrastructures, as they physically or logically connect multiple parts of the same system. Their states are listed in Table 5-6.

*Table 5-6: States associated with connection entities*

| *Connection elements* | *Description* |
|---|---|
| *States* | |
| *First end* | One end of the bidirectional connection. |
| *Second end* | The other end of the bidirectional connection. |

Connections are in their very essence simple concepts. They make use of only two states that are implicitly linked to their existence: a *First-end* and a *Second-end*. Connections are bidirectional and exist to only represent the fact that two entities are tied to each other. Infrastructure nodes and defenders themselves do not use any elements related to connections, but cyberattacks can under the right circumstances make use of connections to infect neighbouring nodes. While connections have little implicit value by themselves, they can be used to specify delays or other elements relating to a specific application if desired. On an ecosystem level however, this is not relevant, as it is not generalisable and would add further assumptions.

*Dependencies*

***Dependencies*** are similar to connections in the way that they represent a form of linkage between infrastructure nodes. However, they are vastly different, as they were described in chapter 2. Connections relate to architectural unity between multiple nodes, whereas dependencies describe the degree of functional impediments across multiple nodes (Setola & Theocharidou, 2016). Crucially, dependencies are not related to a single infrastructural sector, as they can also occur as cross-sectorial dependencies. The core of these links within a cybersecurity ecosystem revolves around impeding dependent nodes in case of inoperability. The states related to dependencies are shown in Table 5-7.

*Table 5-7: States associated with dependency entities*

| *Connection elements* | *Description* |
|---|---|
| *States* | |
| *Origin* | The original node, on which the dependent node relies for operability. |
| *Target* | The dependent node, which depends on the origin of a dependency for their operability. |
| *Weighting* | The weighting associated with a dependency that determines how heavily dependent nodes are impacted if the original node is disrupted. |
| *Current state* | The current state of the original node that determines (along with the weighting) how significantly the dependent node is affected. |

Dependencies incorporate four internal states to store and process information. Dependencies have an *Origin*, or original node, and a *Target*, or dependent node. Since dependencies are directional (unlike connections, which are bidirectional), the direction of a dependency describes the flow of functional dependency (Setola & Theocharidou, 2016). The degree of operability of the origin node

impacts the dependent node, to a certain extent. The extent to which this happens is encapsulated by one further element: the *Weighting* of the dependency. The weighting dictates how severely the impediment to the dependent node will be, given disruptions in the original node (Setola & Theocharidou, 2016). Heavier dependencies can lead to quick disruption of the entire ecosystem, as functional dependencies are widespread (Pederson et al., 2006; Rinaldi et al., 2001). Associated with the level of operability in the original node is the *Current state* of a dependency. This state directly matches the degree of external inoperability it passes on to the dependent node.

### 5.3.3 Specification of objects

The third category of entities to be discussed are *Objects.* As shown in Table 5-1, four such entities are identified. Objects refer to elements that are used by agents to facilitate their interaction, but cannot make autonomous or independent decisions. The objects defined for this conceptual model are *Infrastructure nodes, Defensive strategies, Control mechanisms* and C*yberattacks*.

*Infrastructure nodes*

Infrastructure nodes are the central object within the ecosystem. This is highlighted by their central positioning in Figure 5-1, as all agents exert interaction with infrastructure nodes. Infrastructure nodes, simply put, are the physical systems that are part of critical infrastructures. They are components of a greater network and are directly tied to grave consequences in case of inoperability. Defenders directly control the operation of these nodes and aim to secure the system from undesired intrusions. The main states associated with infrastructure nodes are shown in Table 5-8.

*Table 5-8: States associated with node objects*

| *Node elements* | *Description* |
|---|---|
| *States* | |
| *Internal operability* | The degree of operability posed to a node due to internal affairs, e.g. a successful cyberattack. |
| *External operability* | The degree of operability posed to a node due to external affairs, e.g. an original node impedes this node through dependencies. |
| *Physical impact* | The amount of physical damage that could possibly be sustained by node inoperability. |
| *Economic impact* | The amount of economic damage that could possibly be sustained by node inoperability. |
| *Operation* | The level of operability currently achieved for an infrastructure node. |
| *Current state of operation* | The associated state with the level of operation following Setola and Theocharidou (2016). Possible state are *normal*, *stressed*, *crisis* and *recovery*. |

Infrastructure nodes contain six internal states: *Internal operability, External operability, Physical loss, Economic loss, Operation* and *Current state of operation*. Internal and external operability encompass a similar concept, yet are caused by different entities. The internal degree of operability is the direct result from a defender's capability of defending a node. Successful attacks (false negatives) and non-existing detections (false positives) lead to a decrease in the level of operability present internally. The external degree of operability is the indirect result from other defender's capabilities of defending their respective nodes, which impede this node due to dependencies. Together, these operability components impede a node's level of *Operation*. This is directly connected with a node's *Current state of operation*, which dictates whether losses are being

incurred. Together, these elements describe the extent to which losses are incurred. To establish exactly how much damage is being sustained, *Physical loss* and *Economic loss* describe the possible extent of losses. A design decision was initially made to mirror these states to states used by attackers, as this paves the way for straightforward operationalisation of these concepts. Keeping interfaces clear and simple while maintaining interpretability for states ensures that the model is both manageable and accurate.

*Defensive strategies and control mechanisms*

The second and third objects discussed are *Defensive strategies* and *Control mechanisms*. A defensive strategy is defined as a configuration for a set of control mechanisms. For that reason, the parameters that would describe a defensive strategy are the exact same as for control mechanisms. These two objects are therefore grouped together to avoid convolution of concepts. A control mechanism, as described in chapter 3, is a mechanism that is used to thwart cyberattacks. This can be either through prevention, intrusion detection and responses. The states associated with defensive strategies and control mechanisms are displayed in Table 5-9.

*Table 5-9: States associated with defensive strategy/control mechanism objects*

| *Defensive strategy/control mechanism elements* | *Description* |
|---|---|
| *States* | |
| *Prevention sensitivity* | The likelihood for a prevention mechanism to correctly classify an attack. Derived from the false negative rate. |
| *Prevention specificity* | The likelihood for a prevention mechanism to correctly classify user traffic. Derived from the false positive rate. |
| *Detection sensitivity* | The likelihood for a detection mechanism to correctly detect there is an intrusion. Derived from the false negative rate. |
| *Detection specificity* | The likelihood for a detection mechanism to correctly predict there is no intrusion. Derived from the false positive rate. |
| *Alleviation threshold* | The operability threshold required for a defender to respond with alleviation. |
| *Alleviation duration* | The duration it takes to alleviate intrusions and resume unhindered operation of a node. |
| *Retention threshold* | The operability threshold required for a defender to respond with retention. |
| *Retention duration* | The duration it takes to retain intrusions and restart operation of a node. |

The distinction between preventive, detective and responsive control mechanisms is represented by the states associated with control mechanisms. *Prevention sensitivity* and *Prevention specificity* describe the attributed measures of performance for prevention mechanisms. Similarly, *Detection sensitivity* and *Detection specificity* describe performance metrics for intrusion detection mechanisms. The sensitivity of a control describes the rate at which attacks are correctly classified as such. A failure to do so results in a false negative, as a true attack is not prevented or detected. Specificity denotes the rate at which normal traffic (or a lack of attacks) is classified as harmless and therefore not prevented or detected. If this is not the case, a false positive arises, as legitimate traffic was blocked from the system, or the defender tries to remove an attack that does not truly exist. The other four states directly tie into the perceived operability for a defender. *Alleviation threshold* and *Retention threshold* describe the thresholds for this degree of perceived operability at which a

defender decides to conduct alleviation or retention, respectively. *Alleviation duration* and *Retention duration* describe the time required to remove all attacks and resume normal internal system operation for alleviation and retention respectively. It is important to note that retention always has a heavier impact than alleviation. The threshold for alleviation should therefore always be lower than the threshold for retention. Similarly, because retention is a more drastic measure, it should take less time than alleviation, implying retention duration should always be lower or equal to alleviation duration. The decision to model responses as such is a tentative and assumptious one. There are many ways to model the effectiveness of response mechanisms, such as probability-based clearing of all or some attacks at every step of simulation. This choice is tentative in itself, but a necessary one to establish a conceptual model. The other option would likely not have a significant impact on model behaviour, but that does not mean take away the assumptious nature of the implemented concept.

*Cyberattacks*

The last type of object included in the ecosystem is included in the form of cyberattacks. Cyberattacks are initiated by attackers and are the direct result of their interaction with an infrastructure node. Since a cyberattack is targeted towards a single node, it does not directly interact with defenders themselves. The states applied to a cyberattack object define the way other entities should interact with one. Table 5-10 lists all states associated with cyberattacks.

*Table 5-10: States associated with cyberattack objects*

| *Cyberattack elements* | *Description* |
|---|---|
| *States* | |
| *Detection* | Whether an attack is detected or not. |
| *Power* | The relative power of this cyberattack to the level of operation in a node. |
| *Method of attack* | Whether the attack is conducted as a *Denial-of-Service* or *Worm* attack. DoS attacks are aimed at individual nodes, whereas worms can spread through connections. |
| *Chance of spreading* | Only applicable to worms: If the cyberattack is a worm, the worm can spread to other connected nodes. The worm arrives as if it were regular traffic, and is therefore first subjected to prevention mechanisms. |
| *Duration* | How long an attack has lasted. |

There are five internal states that define how a cyberattack interacts with other entities in the system. The first, *Detection*, is a simple check as to whether a control mechanism detected the attack during its lifespan. The second, *Power*, is the degree of internal damage that can be inflicted to a node's operability. The degree of power depends on the type of attack defined in chapter 3. Similarly, the *Method of attack* describes the MO behind an attack, which is either a worm-based attack or a Denial-of-Service-based attack. The *Duration* of an attack is used to keep track of how long attacks have been going on, establishing a performance metric to assess the timeliness of defensive decisions. Additionally, worms can spread through connected neighbouring nodes to further infect the ecosystem. The likelihood of this occurring is given by *Chance of spreading*. These states define a high-level, abstract representation of cyberattacks and their impact on nodes. It is not likely that such a definition could translate directly into real-world cases. These assumptions impact model interpretability significantly, as real-world cases that inspired this study and introduced the knowledge gap have already shown that attacks can be incredibly sophisticated and operate on more

layers than presented here (Department of Homeland Security, 2015; Fairley, 2016; Farwell & Rohozinski, 2011; Liang et al., 2017).

## 5.4 Model performance metrics

This section will briefly discuss which metrics represent model performance appropriately. Before model formalisation steps are conducted, it should be clear where exactly emergent behaviour should be expected. An agent-based model can quickly contain dozens or hundreds of parameters or possible combinations of parameters, many of which provide little insight into dynamic emergent behaviour. Factors that are likely part of the model but will provide no insight in model behaviour include the number of nodes, the number of attackers or average dependency weighting. Instead, changeable, emergent properties should be monitored.

### 5.4.1 Damage to nodes

The first metric to track dynamic model performance is the degree of damage being inflicted to nodes. This includes the total amount of losses incurred, as well as the implicit physical and economic components. The main purpose for the agent-based model is to assess the effectiveness of coherent, top-down defensive strategies in a bottom-up, emergent environment. The severity of consequences forms one of the primary inputs to conduct this study in the first place. Exploring changes to the sustained extent of damage is the most straightforward practice when assessing whether desired behaviour emerges. Since damage can be tracked cumulatively, this shows the relative growth of losses following deviations in other performance indicators for a single model run. Other performance metrics are likely more chaotic in nature, as there is no cumulative growth or improvement in behaviour. Instead, the model is used to assess the system performance under certain parameter configurations, which is likely more chaotic and reactionary in itself. Another similar indication for the system performance within the model is keeping track of the state of operation of nodes in the model.

### 5.4.2 Correctness of defensive decisions

The second metric applicable to track model performance relates to the quality or correctness of defensive decisions. Given the prescribed *Alleviation threshold* and *Retention threshold*, defenders should only conduct decisions when their *Perceived operation* reaches these threshold. However, the true level of *Operation* in a node might vary significantly. By checking the values or deviation among these values, the effectiveness of system configurations can be more accurately explored. This helps map the type of decisions being made across a model parameter configuration. Part of assessing the correctness of defensive decisions involves establishing whether defenders overestimated, underestimated or correctly estimated the level of operation in a node. This can help explain whether losses are primarily incurred due to failing to mitigate cyberattacks or due to intrusion prevention and detection mechanisms raising false alarms. Since the nature of this study is exploratory, this helps account for possible emergent tendencies, since output statistics do not provide relevant insights on their own.

### 5.4.3 Cyberattack surreptitiousness

The third metric used to assess model behaviour is the success rate of cyberattacks and all factors required to establish the effectiveness of cyberattacks. A key element of ecosystem interaction, cyberattacks disrupt functional operation of infrastructures. There are several facets of cyberattacks that can be tracked to observe emergent patterns in model behaviour. Assessing whether cyberattacks manage to bypass prevention mechanisms for nodes in itself does not yield any significant information, as this is inherently tied to the *Prevention sensitivity* state discussed in sub-section 5.3.3. More interesting, however, is assessing how long cyberattacks last before they are

removed and the fraction of attacks that have been detected. Together, the three metrics identified shape a story about what happens over the course of a simulation.

## 5.5 Intermediate findings

This chapter served to establish a conceptual model that incorporates all aggregated model concepts discussed throughout chapters 2, 3 and 4. This required specification of all elements to be incorporated in the simulation model. The first step, discussed in section 5.1, was to identify the framework for model conceptualisation. The complex adaptive systems perspective enables identification of states, actions and interactions on the level of individual agents. The next step, detailed in section 5.2 was to denote the ecosystem of critical infrastructures from the perspective of complex adaptive systems. The translation of the ecosystem into a conceptual model to be used for agent-based modelling was discussed in section 5.3. This was done through creating a model inventory and specifying all key entities in the ecosystem model. It is important to remember the assumptious nature of any such model, as well as the implications of the inclusion of high-level and abstract definitions and interaction. The last step, detailed in section 5.4, was to identify possible metrics for model performance that can be used to observe emergent patterns. Metrics identified relate to the level of operation for nodes, the correctness of defensive decisions made and the success factors for cyberattacks. The next steps in this study will continue with the conceptual model, formalising and implementing the identified concepts.

# 6 Formalising an agent-based model

This chapter entails the model formalisation and implementation process. Building upon the conceptual model established in chapter 5, the following steps relate to translating actions, states and rules towards computable expressions to be included in a simulation model. Following the ABM cycle by Nikolic et al. (2013), the steps taken in this chapter correspond with *Concept formalisation, Model formalisation, Software implementation* and *Model verification*. Conducting these steps properly ensures a robust and representative model to discern emergent patterns in system behaviour. Section 6.1 will first establish deviations made from the conceptual model to the formalised version. This is followed by laying out the model narrative and order of interactions in section 6.2. The key mechanisms required to define the actions discussed in section 6.1 are discussed in section 6.3. Subsequently, section 6.4 details the process of implementing the model and how the model can be used. Next, model verification processes are described in section 6.5. Finally, section 6.6 wraps up this chapter and defines key concepts identified along the way.

## 6.1 Deviations from conceptual model

To reduce the degree of complexity involved with the modelling process, several deviations were made during the formalisation process. These steps do not further abstract complex concepts into simpler computations, but were applied to save resources for eventual simulation (Martin, 2009). Martin (2009) perceives the codification process of complex systems as an incremental process, where system elements are designed as a minimal model and expanded upon with new concepts that could improve the system architecture.

The first change made is to not implement user agents as ABM agent entities at all. Given the relative simplicity of their states and actions, their interactions with infrastructure nodes are modelled as implicit actions undertaken by infrastructure node operators. Every action and state conceptualised for user agents are still included, modelled as elements belonging to infrastructure nodes.

The second change made is to not implement infrastructure operators and infrastructure nodes as separate entities. Instead, defender agents represent both the node object and the original defender agent. This does not change the way interactions take place, as defenders only interacted with other entities through infrastructure nodes. Furthermore, defenders were originally already tied to a single node. This results in defenders interacting with other entities directly, as opposed to indirect interaction, which could prove more problematic for modelling purposes. By doing this, only the essential agents that display independent, autonomous decision-making are modelled as such. All other entities provide the exact same data they would otherwise do, without convoluting the simulation model with unnecessary agents. This is done to avoid convolution of procedures within the model, as related concerns are grouped, and separate concerns are decoupled at the code level (Martin, 2009).

The third and last change made is to implement cyberattacks as a link entity. This is done for two reasons. Cyberattacks as objects need to be capable of spreading, which is done easiest if they are capable of calling another procedure. The other reason is that a link shows the active attack from an attacker to a node more clearly. Especially when the attack is worm-based and has spread to multiple nodes, this can more easily be distinguished from the rest of the model.

## 6.2 Model narrative

This section discusses the narrative of model interaction. In order to incorporate all actions and interactions discussed in chapter 5, the model includes twelve main procedures, thirteen if setup procedures are included. To illustrate the overall structure of procedures, Figure 6-1 depicts the

structure of overall procedures. A distinction is made between procedures initiated by attackers (highlighted in blue) and those initiated by defenders (highlighted in green). The flowchart shows how iteration at each distinct time step takes place, aside from the initialisation process, which is only conducted at model setup. Although attacker procedures flow into defender procedures, these can be seen as separate entities. The former will be discussed first, followed by the latter.

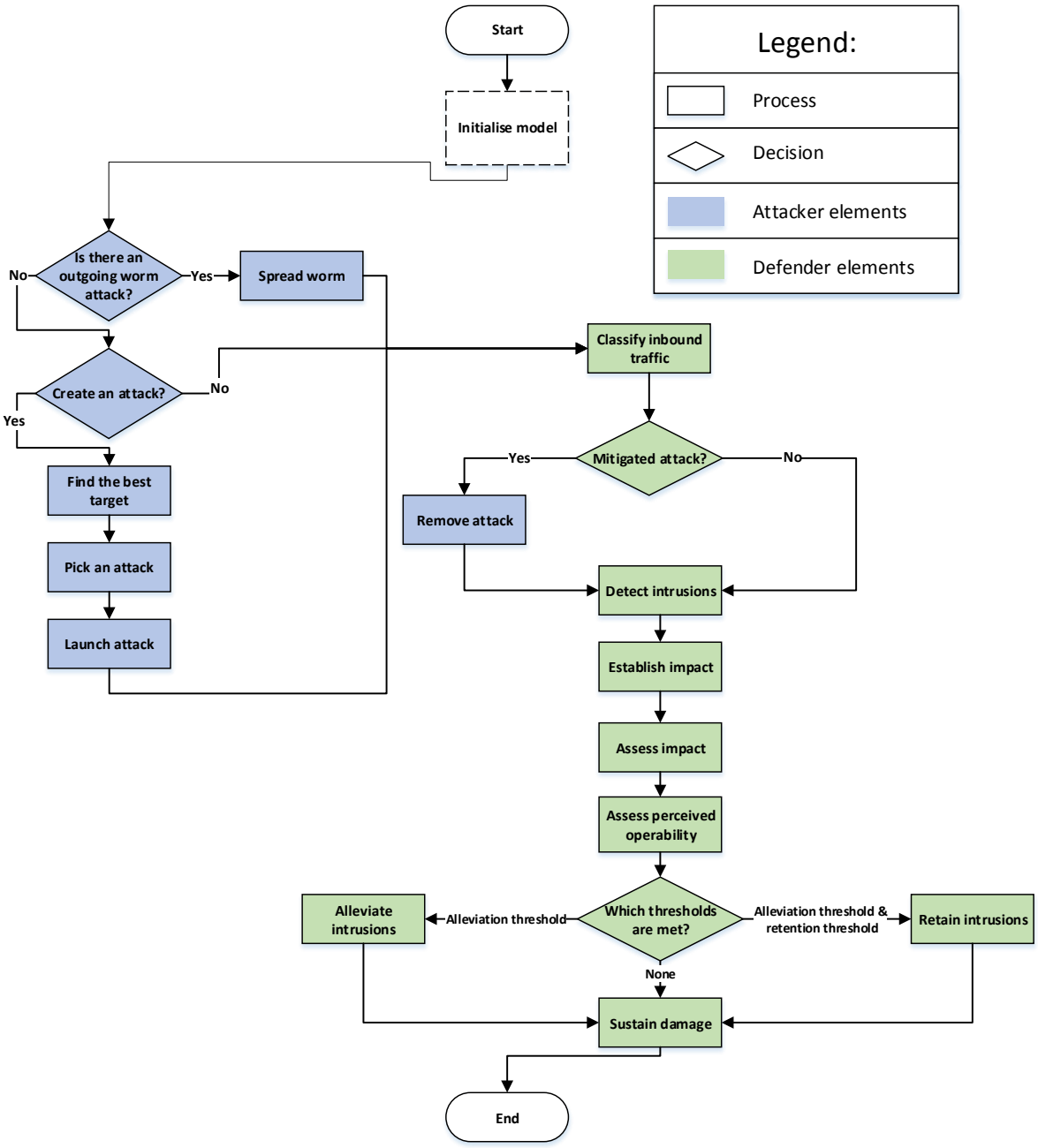


*Figure 6-1: Flowchart depicting the structure of model procedures*

### 6.2.1 Attacker procedures

The first step taken by attackers is to assess whether they should attack. Attackers can only launch a new attack if they are not currently attacking. If an attacker is already attacking, they will then assess whether this attack is a worm. If they are conducting a worm-based attack, they will assess whether the worm should spread further or not. Worm-based attacks have a chance of spreading to connected nodes of the target. There is a predetermined chance of this happening at each time

interval. If the condition to spread worms is met, the attacker determines which connected neighbour a node will spread the worm to. The worm then spreads as if a new attack was launched towards the neighbouring node. If this is not accounted for, the worm can quickly spread across large parts of the network.

If attackers are not going to initiate a new attack, their procedure ends and another agent will be iterated over. If an attacker is not attacking and they do decide to initiate a new attack, they need to first identify several elements. The first is the target infrastructure node. Attackers select a target that corresponds with their personal preference for type of damage sustained, as well as the overall extent of damage sustained. The degree of knowledge available to an attacker determines the level of detail to which they pick their target. Attackers with low knowledge pick their targets randomly, and attackers with higher levels of knowledge are capable of computing internal damage expectations and possible cascading failures.

The next step is to identify which attack would fit best. To this end, attackers simply pick the most powerful attack. The choice of attack is logically limited to whether a certain attacker is capable of conducting each type of attack.

Before an attacker actually creates an attack instance, they need to pass a last check: whether their perceived attack utility assessment outweighs the degree of utility they could experience by attacking targets outside of the ecosystem. If this is the case, they will be considered as actively attacking, while not creating any attack entities. This automatically implies that more knowledgeable attackers are also more likely to attack, since they have full knowledge on any and all first-order dependent nodes being impacted.

### 6.2.2 Defender procedures

The defender procedure starts with checking whether there is any inbound traffic. This traffic is classified following the definitions of prevention mechanisms discussed in section 5.3. The classification of traffic works differently for inbound attacks and user traffic. Since user traffic is modelled implicitly, a defender calls this procedure themselves. If an attack is classified correctly, the defender calls on the associated attacker to remove the attack. If an attack is classified incorrectly, it remains intact and undetected. If user traffic is erroneously prevented, the internal operability of the associated node is tarnished for a set amount of time.

The next procedure is also related to the implementation and configuration of control mechanisms, being the detection of intrusions. Prevention mechanisms are activated by any traffic being initiated, whereas detection mechanisms are applied at each time step. Defenders assess at each time step whether there are any inbound attacks, along with the associated impact they expect. This also yields the chance to throw up false positives and false negatives. Failure to detect an attack results in overestimated perceived operability of a node, whereas false positives result in underestimated perceived operability. These elements both tie into the impact assessment and perception procedures.

Following the prevention and detection procedures, a node conducts two procedures related to their impact assessment. The first procedure serves to establish the internal and external impact components and to derive the actual level of operability. The second procedure represents the situational awareness a defender acts upon and establishes the perceived extent of impact. Undetected intrusions and erroneously prevented user traffic both contribute to the true level of inoperability without being included in the perceived impact assessment.

Conversely, false positives resulting from erroneous intrusion detection are not included in the impact assessment, as non-existing attacks do not inflict damage directly, while contributing to a false degree of situational awareness. This is the direct result of limited and incorrect information available to a defender. In terms of the external impact components, these are both the direct result of operability of origin nodes with dependencies towards a node. There is no difference between the perception and true degree of external inoperability, as node operators can discern when a lack of productivity is transferred as input for their internal operability.

Based on the perceived operability, which is the direct result of perceived internal and external impact awareness, defenders select the most appropriate response mechanism. After establishing the degree of operability based on internal and external impact perceptions, this is compared with two thresholds. Given the shared, top-down defensive strategy, these thresholds are universal across all infrastructure nodes. If no threshold is met, no response is executed. If only the alleviation threshold is met, the defender will start alleviating intrusions within the node. This keeps the node in operation and is the safe, risk averse approach for overall damage sustained. However, keeping the node in operation allows worm attacks to spread to connected neighbours. On the other hand, intrusion retention is conducted if the more stringent threshold for retention is surpassed. This closes the node in order to ensure attacks inflict no further damage in the ecosystem. This takes a shorter time than alleviating intrusions, but means that full internal losses are experienced during the duration.

The last defender procedure relates to the degree of operation within every node and the associated losses. Physical and economic losses are experienced at each time interval. The extent of damage is inversely based on the operability level of a node, multiplied by the total possible losses associated.

## 6.3 Model specification

This section will tread into further detail as to how certain core procedures are formalised. To achieve this, the concept formalisation will be discussed first. This is followed by the formalisms applied to core model mechanisms. Lastly, the time scale to which the model is tailored will be detailed.

### 6.3.1 Concept formalisation

The first step in formalising the conceptual model into a fully-fledged simulation model is to identify which states are to be included as software elements. Nikolic et al. (2013) assess the importance of this step as it helps establish whether formerly logical elements might have been more context dependent than expected. Appendix B denotes full details as to which elements are included, to which category these belong and how these software types are formalised. Appendix C shows flowcharts that depict all procedures to be found in the model. These elements largely match the state definitions discussed in chapter 5, having applied the deviations discussed in section 6.1. Whilst there are several additional deviations, these are all very limited in impact and the purpose for this becomes clear in section 6.4.

### 6.3.2 Model formalisms

The next step is to formalise all conceptualised model interaction. There are multiple approaches to this step, including UML diagrams, pseudo-code and flowcharts (Nikolic et al., 2013). For this study, flowcharts are used, similar to the description of the model narrative. Given the deep nestedness of interaction among agents, flowcharts serve as a great visual aid to discern between agent behavioural inputs and outputs. Additionally, there are several core mechanisms central to model interaction. These operationalisation of these mechanisms will be discussed throughout this sub-section.

*True degree of inoperability*

As stated in chapter 5, the level of operation is computed through two components of impact or inoperability. Inoperability is measured as a continuous scale between 0 and 1 and is the inverse of the level of operation. Since both the degree of inoperability and operation are modelled as scales from 0 to 1, the level of operation $O$ can be derived by the internally limited level of operation and the externally limited level of operation. These limited levels of operation are derived from the internal impact $\rho_{internal}$ and external impact $\rho_{external}$ as defined in (1).

$$O = (1 - \rho_{internal}) \times (1 - \rho_{external}) \tag{1}$$

$\rho_{internal}$ is established by the sum of the impact of all inbound attacks (*A*), as well as additional impact attributed to the sum of the impact of blocked users (*U*). The internal inoperability component is given by (3).

$$\rho_{internal} = \begin{cases} \sum_{A_i \in A} I(A_i) + \sum_{U_j \in U} C(U_j), & if \sum_{A_i \in A} I(A_i) + \sum_{U_j \in U} C(U_j) \leq 1 \\ 1, & if \sum_{A_i \in A} I(A_i) + \sum_{U_j \in U} C(U_j) > 1 \end{cases} \tag{2}$$

With:

- $A_i$: an attack from the set of all active attacks *A* towards this node
- $U_j$: a user from the set of all active users currently erroneously blocked from this node
- *I*: the power or impact associated with an attack
- *C*: the criticality of user traffic

$\rho_{external}$ is established by dependencies, following the hampered level of operation in all origin nodes. By extent, this refers to the level of inoperability $(1 - O)$ in each node. The following equation denotes this relationship:

$$\rho_{external} = \begin{cases} \sum_{D_k \in D} w_{D_k} \times (1 - O_k), & if \sum_{D_k \in D} w_{D_k} \times (1 - O_k) \leq 1 \\ 1, & if \sum_{D_k \in D} w_{D_k} \times (1 - O_k) > 1 \end{cases} \tag{3}$$

With:

- $D_k$: the dependency from origin node *k* out of the set of all inbound dependencies *D*
- $w_{D_k}$: the weighting of dependency $D_k$
- $O_k$: the level of operability for origin node *k*

*Impact assessment*

The calculation of perceived operability is almost identical to deriving true inoperability. However, there are slight differences resulting from the implications of situational awareness. The perceived degree of operability $p_{operability}$ is computed identically to the true degree of operability, albeit based on the respective perceived components.

$$p_{operability} = (1 - p_{internal}) \times (1 - p_{external}) \tag{4}$$

With:

- $p_{operability}$: the perceived degree of operation

- $p_{internal}$: the perceived value for $\rho_{internal}$
- $p_{external}$: the perceived value for $\rho_{external}$

However, the difference between an actual situation and a defender's situational awareness emerges in how internal inoperability is computed. While the actual impact is composed by all attacks, perceived impact is composed by only detected attacks. Additionally, defenders cannot tell whether they erroneously blocked legitimate traffic, leading to that facet of internal impact missing. On the other hand, false positives in intrusion detection lead a defender to believe the situation is more problematic than it really is. False positives for intrusion detection and prevention are modelled to expire the next time step, when the error is overturned. The equation for $p_{internal}$ is as follows:

$$p_{internal} = \begin{cases} \sum_{A_i \in A_{detected}} I(A_i) + \sum_{FP_j \in FP} I(FP_j), & if \sum_{A_i \in A_{detected}} I(A_i) + \sum_{FP_j \in FP} I(FP_j) \leq 1 \\ 1, & if \sum_{A_i \in A_{detected}} I(A_i) + \sum_{FP_j \in FP} I(FP_j) > 1 \end{cases} \quad (5)$$

With:

- $A_i$: an attack from the set of all active detected attacks $A_{detected}$ towards this node
- $FP_j$: a currently detected false positive from the set of falsely detected, non-existent attacks on this node
- *I*: the power or impact associated with an attack

The external inoperability component is no different for the degree of perceived impact than it is for the true extent of inoperability. Defenders are able to tell when origin nodes are not delivering the performance dependent nodes expect, putting instantaneous pressure on node operation. The external component is thus computed exactly the same way as equation (3).

$$p_{external} = \rho_{external} \quad (6)$$

*Intrusion prevention and detection*

Another key mechanism is the prevention and detection of intrusions by defenders. As opposed to the computation of risk assessment and perception, these processes do not involve any sophisticated computations. Figure 6-2 depicts the process and order of operation formalised for intrusion prevention. Intrusion prevention can be called for two types of events: an attacker attempting to launch an attack and a defender simulating user traffic reaching the node. Both events call for classification into either attacks or harmless, legitimate traffic. Incorrect classifications result in an underestimation for perceived risk and an increase in true risk exposure. The success of these classifications is based on the effectiveness of the prevention mechanism, which is the expressed by prevention sensitivity and prevention specificity. Random values between 0 and 1 are generated to assess whether a successful event occurs, with prevention sensitivity and prevention specificity each denoting the success rate of classification.

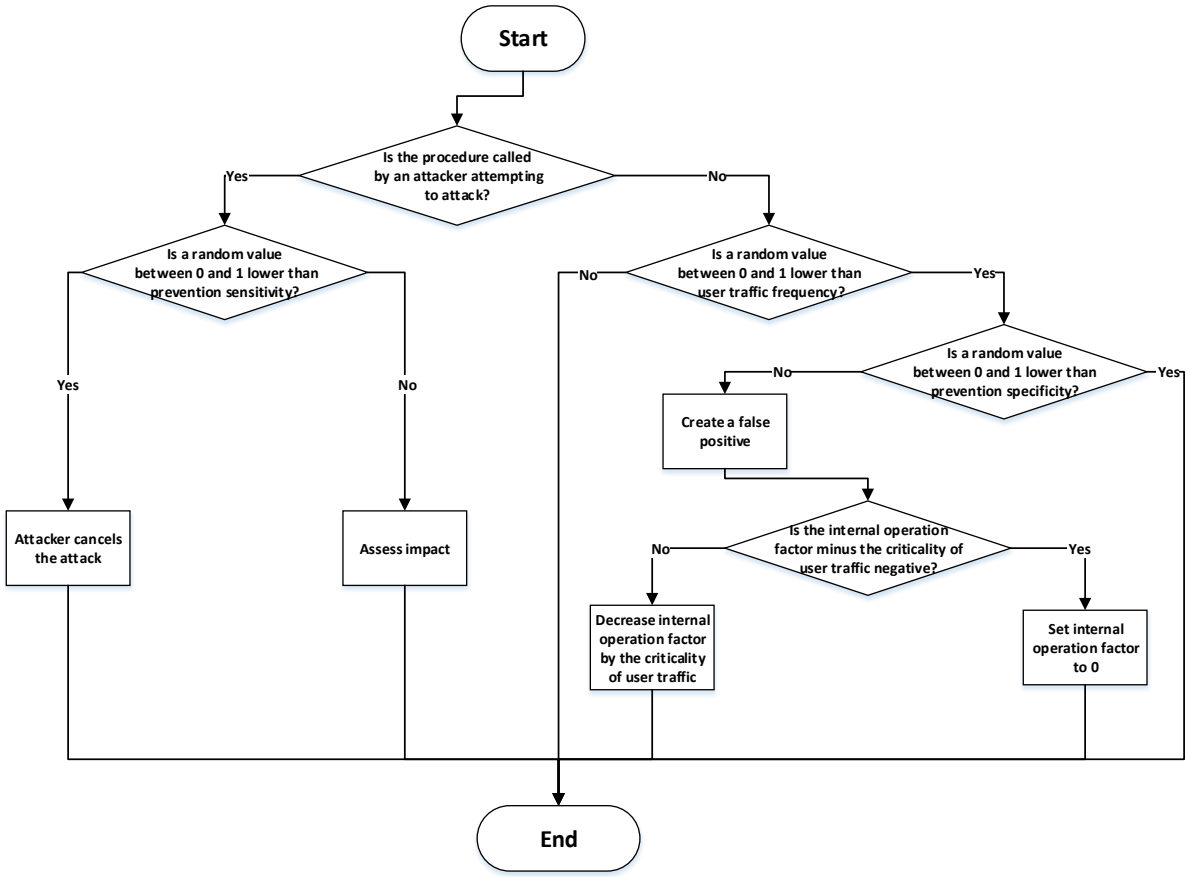


*Figure 6-2: Flowchart for intrusion prevention procedures*

The procedure for intrusion detection also consists of two separate sub-processes. The first deals with possibly generating false positives, which in turn leads to an overestimation of the risk perception. The second sub-process deals with detecting existing attacks. Failing to appropriately classify an attack lead to no changes being made, as the undetected attack already contributes to the true degree of risk exposure. Detecting an intrusion correctly helps bring the degree of perceived impact more in line with actual effective impact and helps feed into eventual response decisions as discussed in section 6.2. The intrusion detection process is shown in Figure 6-3. The success of these classifications is based on the effectiveness of the detection mechanism, which is the expressed by detection sensitivity and detection specificity. Since detection sensitivity and detection specificity both represent the success rate of classification of attacks and non-attacks, random values between 0 and 1 are used to determine whether classification results in a true positive, false positive, true negative or false negative classification.

*Damage calculation*

The procedure of damage computation relies on two main factors as discussed in chapter 2: physical and economic losses. The total degree of losses $L_{total}$ is given as the sum of these components (7). Both components are in turn computed by the product of the extent of inoperability and the associated respective loss factors $F_{physical}$ or $F_{economic}$. These computations are shown in (8)and (9).

$$L_{total} = L_{physical} + L_{economic} \quad (7)$$

$$L_{physical} = (1 - O_{overall}) \times F_{physical} \quad (8)$$

$$L_{economic} = (1 - O_{overall}) \times F_{economic} \quad (9)$$

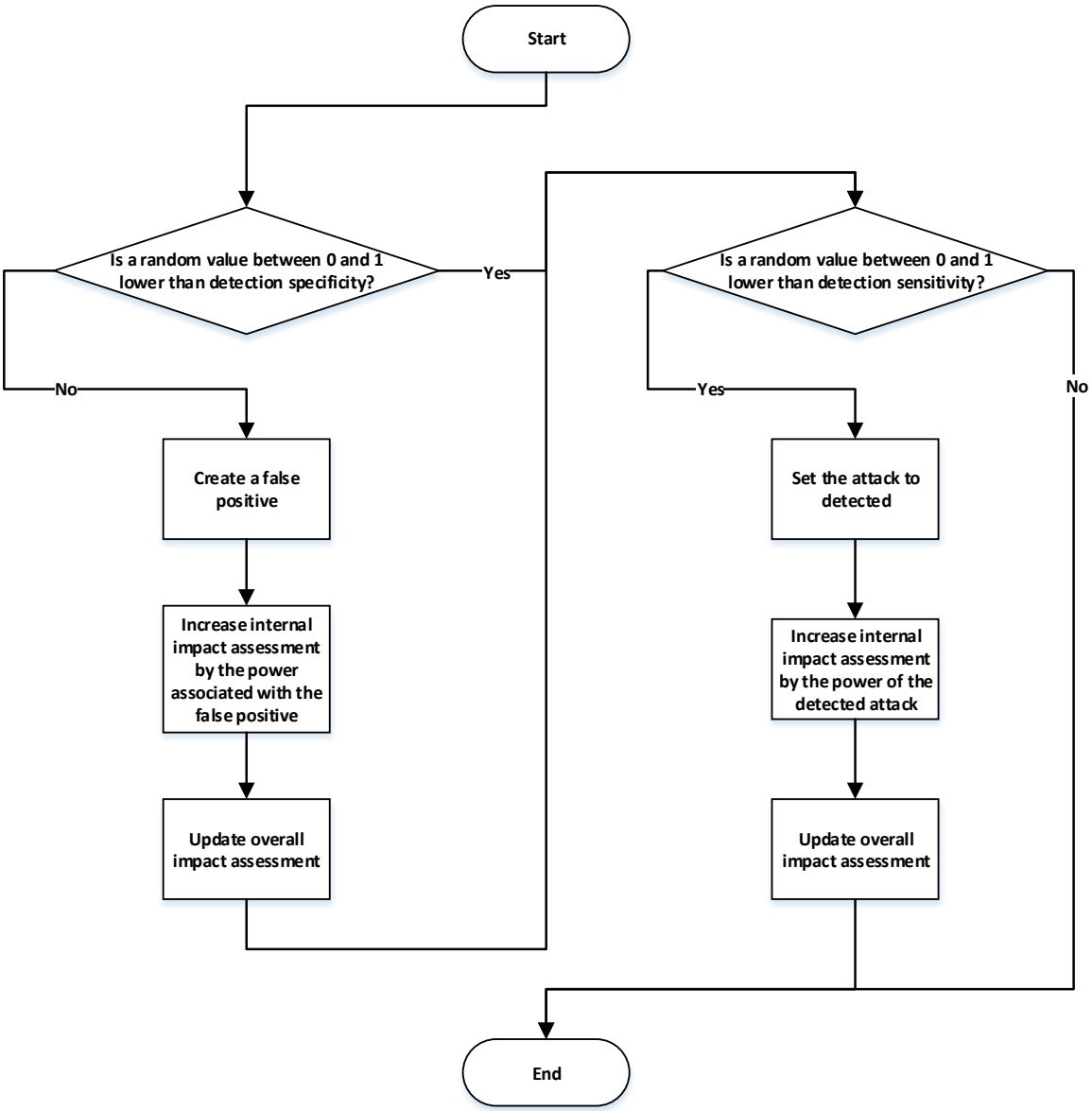


*Figure 6-3: Flowchart for intrusion detection procedures*

*Target and attack selection*

The mechanism used to select a target depends on the overall knowledge level $K$. The cases for each value of $K$are shown in (10). Indicative of attacker interaction is the depth of their perception growing as their knowledge level increases. Being able to assess weighted loss more clearly yields a more refined target selection scheme, which increases the capabilities of attackers as well as their likelihood to attack nodes within the ecosystem. The lowest level of knowledge, shown in (11), randomises the attacker from the target population $T_i$. The medium level of knowledge, shown in (12), maximises the perceived attacker utility from attacking the isolated target alone. The highest level of knowledge, shown in (13), maximises perceived attacker utility for attacking each target as

well as all outgoing dependent nodes. This is given by the maximisation of attacker utility for the target node added to all attacker utility attained from first-order dependencies.

$$T = \begin{cases} Equation\ (11), if\ \ K = "low" \\ Equation\ (12), if\ K = "medium" \\ Equation\ (13), if\ \ K = "high" \end{cases} \tag{10}$$

$$T = Random(T_i) \tag{11}$$

$$T = \max_{T_i}( F^i_{physical} \times P_{physical} + F^i_{economic} \times P_{economic}) \tag{12}$$

$$T = \max_{T_i}( \left(F^i_{physical} \times P_{physical} + F^i_{economic} \times P_{economic}\right) + \sum_{D_j \in D_{out}} w_{Dj} \times \left(F^j_{physical} \times P_{physical} + F^j_{economic} \times P_{economic}\right)) \tag{13}$$

With:

- $T$: the selected target
- $T_i$: A target $i$ from the set of all targets available (nodes who are not in crisis or recovery)
- $F^i_{physical}$: factor of physical loss associated with a node $i$
- $F^i_{economic}$: factor of economic loss associated with a node $i$
- $P_{physical}$: attacker preference for physical damage
- $P_{economic}$: attacker preference for economic damage
- $D_{out}$: the set of outgoing dependencies
- $D_j$: A dependency $j$ from the set of outgoing dependencies $D_{out}$ leading to a dependent node associated with this dependency $j$
- $w_{Di}$: the weighting of a dependency $D_j$

The mechanism applied to assess which attack to use is relatively straightforward: attackers with all levels of knowledge pick the most powerful attack. Since attack powers are not context-dependent, it would be unrealistic to assume attackers have no knowledge over their own resources. The equation for this maximisation problem is shown in (14).

$$\alpha = \max_{a_i \in \alpha_{capable}} (I(a_i)) \tag{14}$$

With:

- $\alpha$: the selected type of attack
- $a_i$: an attack type $i$ iterated over
- $\alpha_{capable}$: the types of attack available to this attacker
- $I(a_i)$: the impact or power associated with an attack of type $a_i$

### 6.3.3 Time scale

The time scale the model will iterate over is based on a time interval of one day, meaning one tick in the simulation model corresponds with one day. Because choices made during model formalisation reduced the computational strain exerted by the simulation model, thousands of ticks can be

iterated through relatively quickly. A daily time interval allows for threats to occur following more realistic distributions, while avoiding to create a model that is extremely sensitive to very uncommon events. Given the aim of this study to assess the effectiveness of different coherent defensive strategies, a time interval that stretches over multiple years is applied. However, to avoid dealing with uncertainties with regards to developments in the ecosystem, the overall timespan should be constrained. With these requirements in mind, a timespan of five years (or 1825 days) was chosen, as this allows for robust assessment of defensive strategies without straying too far into the unknown.

## 6.4 Software implementation

This section discusses elements related to the software implementation phase. First, the chosen software package will be discussed. Next, an overview of what model elements look like is given.

### 6.4.1 NetLogo

The software package chosen to implement this agent-based model is NetLogo for Windows, version 6.0.2. NetLogo, developed by Northwestern University's Uri Wilensky, offers an easy-to-understand software environment that allows researchers to model complex adaptive systems (Tisue & Wilensky, 2004). NetLogo is particularly well-suited for studies that assess the behaviour of different system configurations. Since this study is not primarily quantitative, NetLogo offers a suitable and applicable software environment to implement the formalised model for thorough exploration and experimentation (Nikolic et al., 2013). The true representativeness of the model is not to compare real-world policy instruments, as those would require incredibly specific and non-generalisable models, but instead to explore what happens if top-down defensive strategies were applied to different scenarios.

### 6.4.2 Model overview

The simulation model now implemented in NetLogo features several elements that require further explanation. The overview of the model is shown in Figure 6-4. There are four distinct elements of the model interface: *model input parameters*, *buttons*, *the model* and *output monitors.*

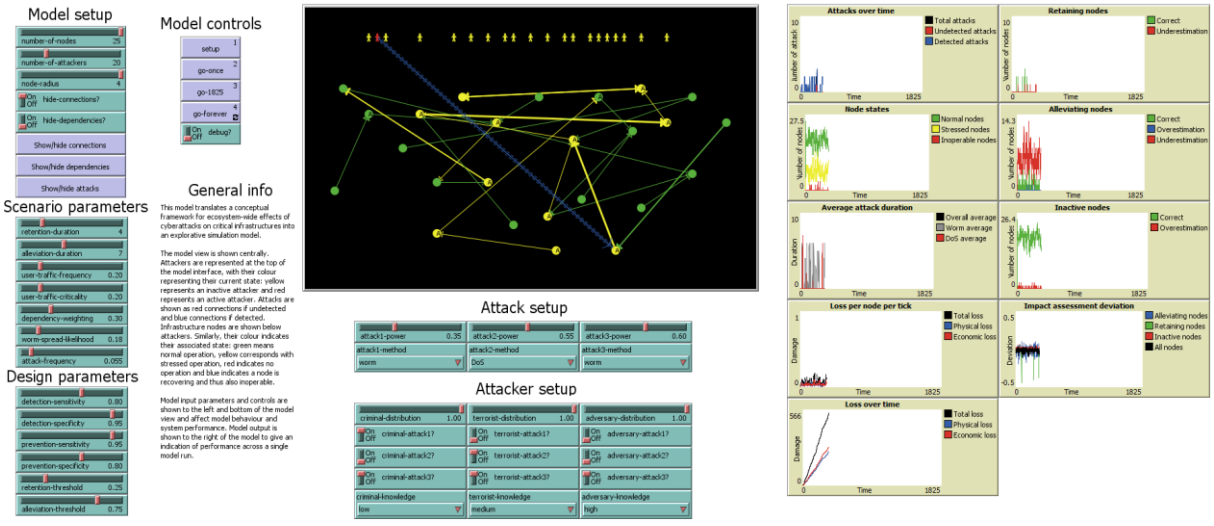


*Figure 6-4: Model overview in NetLogo*

Model input parameters, such as sliders, choosers and switches, prescribe values used during model usage. An example of each is shown in Figure 6-5. Model input parameters can be altered to affect core functionality of the model. A division was made between *attack setup*, *attacker setup*, *model setup, scenario parameters*, and *design parameters.* The first two types are all supposed to be static for experimentation, but can be altered over to explore model behaviour for a baseline

configuration. The model setup parameters are purely in place to make the model appear less convoluted. Scenario parameters are uncertain factors that are varied over to suppress uncertainty. The last set, design parameters, serve as the foundation for experimentation.

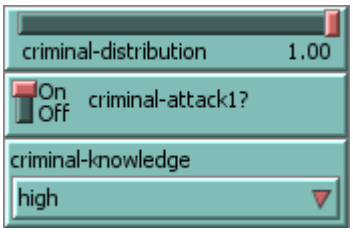


*Figure 6-5: Examples of model input parameters. From top to bottom: slider, switch, and chooser.*

Buttons are used to operate the model. The buttons used in the model are shown in Figure 6-6. Buttons directly operate procedures to be called. There are two main operating procedures: *setup* initiates the model and clears digital residue from previous model runs and *go* operates the main system procedure, which in turn calls on all sub-procedures. There are three types of *go* buttons: go-once simulates one tick, go-1825 simulates 1825 ticks and go-forever continuously simulates ticks until the button is depressed. The other buttons are used to show or hide elements that might be convoluting the model view.

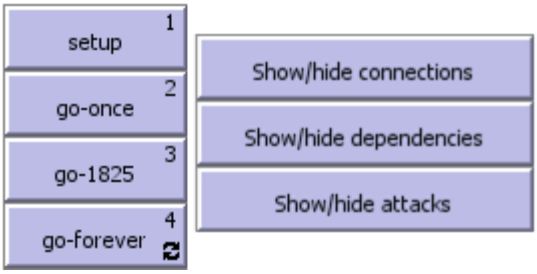


*Figure 6-6: Buttons in NetLogo*

The model view is the main element of NetLogo's interface. The model view depicts all elements of interactions that elicit visual changes. Two examples are shown below: Figure 6-7 depicts the model view if no elements are hidden, whereas Figure 6-8 shows the default model configuration, hiding connections. Visual elements shown are:

- Attackers (the top row of entities) with their state (*attacking?*) determining their display colour: red attackers are currently attacking, whereas yellow attackers are not.
- Infrastructure nodes or defenders (dots in the middle of the model view) with their state (current-state) determining their display colour. Green nodes are in normal operation, yellow nodes are stressed, red nodes are inoperable and blue nodes are recovering. The label "A" indicates alleviation is in process and the label "R" indicates retention is in process.
- Attacks (connections between attackers and nodes) with their shape affected by the associated attack method and their colour set by their state (detected?). A blue attack is detected, a red attack is not.
- Dependencies are directed links between two nodes. The line thickness is affected by the *weighting* and their colour matches the state of the origin node (except for recovering nodes which exert a red dependency).
- Connections are undirected links between two nodes. They have no further properties.

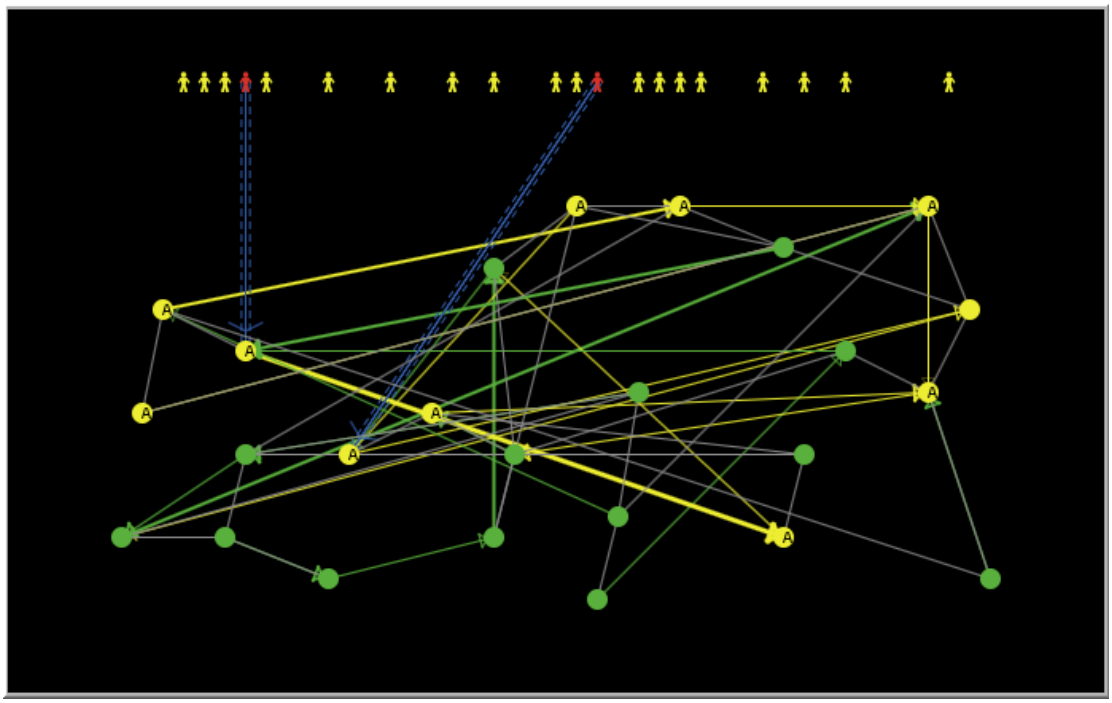

*Figure 6-7: Model view in NetLogo. Hiding no elements.*

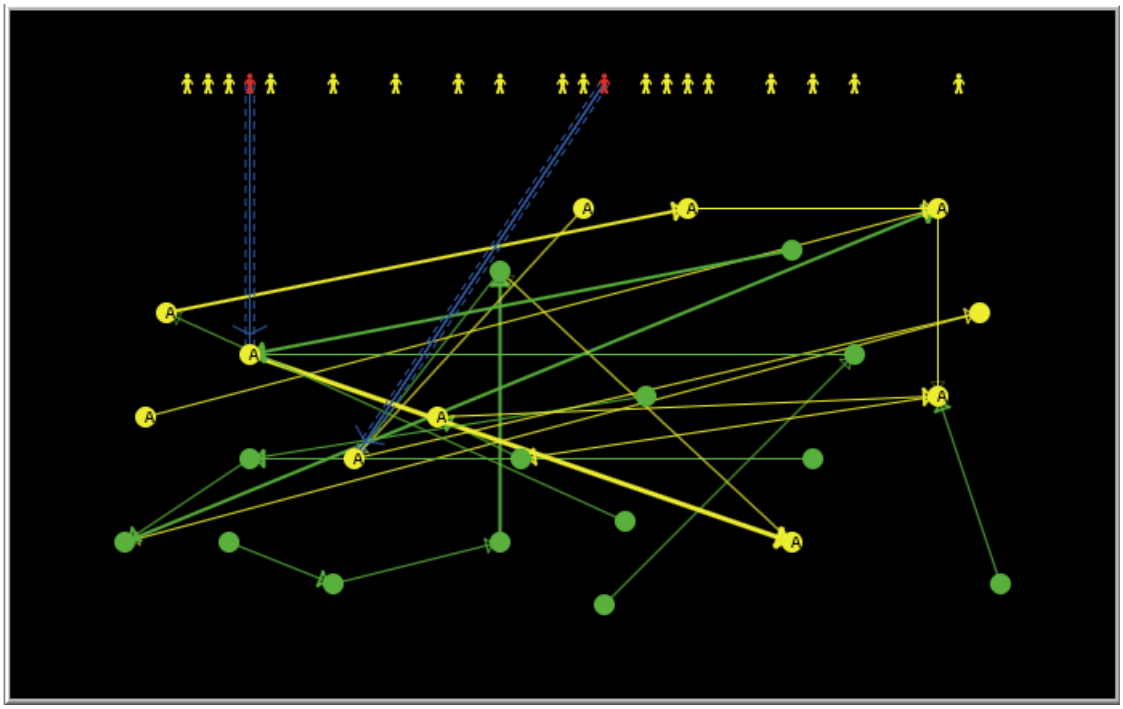

*Figure 6-8: Model view in NetLogo. Hiding only connections.*

The last element of the model interface are output monitors. Only one type of output monitors are used, *plots*. Plots indicate the development of certain model parameters over time. Each plot can include multiple graphs and can be used to indicate differences in model performance across different sets of agents. Two examples are shown in Figure 6-9.

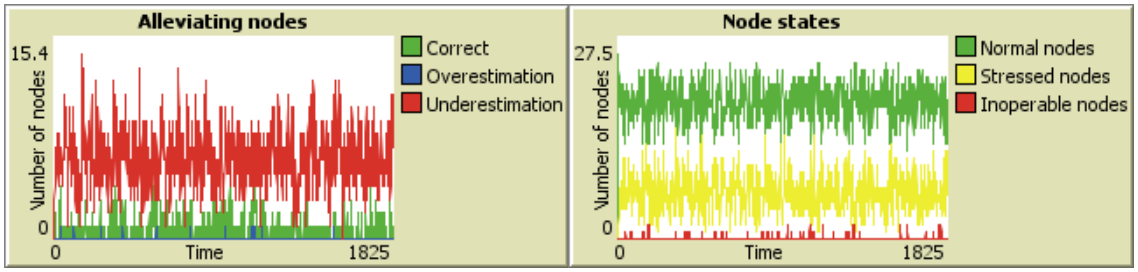


*Figure 6-9: Model output plots*

## 6.5 Model verification

After implementing an agent-based model, the model should be verified to ensure that the way the model operates is in line with desired interactions and that no errors were made during implementation (Nikolic et al., 2013). When designing a model that might be difficult to validate, which is typical for ecosystem-level simulation models, extra emphasis should be placed on the validity of concepts implemented in the model (Augusiak et al., 2014; Kwakkel & Pruyt, 2013). Martin (2009) prescribes thorough unit testing for any coding projects, as the only meaningful elements of code are those that have been subjected to clean testing. The *Evaludation* framework put forth by Augusiak et al. (2014) proposes an integrated strategy for validating models when parameter settings are difficult to quantify or substantiate. As part of this framework, thorough verification of model implementation and output is required, but exact pointers to each step are not specified. The emphasis placed on verification by the *Evaludation* framework corresponds with the agent-based modelling cycle by Nikolic et al. (2013). For this reason, verification is conducted following the agent-based modelling cycle. Model verification is conducted in four steps:

1. Tracking agent behaviour: actively implementing debug procedures during model implementation to ensure that the desired result was achieved for each implemented set of commands.
2. Single-agent testing: testing whether values are computed correctly by a single agent for each type of agents, ensuring that the mechanisms by which individual agents operate are implemented correctly.
3. Interaction testing in a minimal model: testing whether the order and execution of interaction works correctly by implementing a model with the minimal number of entities. This serves as another check to ensure that overall interaction is the desired result from single-agent actions.
4. Multi-agent testing: testing whether values resulting from simulation model repetitions exert the desired variability to ensure values are computed correctly and outliers can be explained. Besides testing the variability, timeline sanity is evaluated for several model repetitions to ensure that desired emergent behaviour results from a set of preconditions.

These four steps were conducted thoroughly and resulted in several implementation errors emerging. The results of these steps are discussed in full detail in Appendix E. By ensuring agent behaviour was consistently tracked during model implementation, the burden on further verification steps was reduced. Debugging tools were continuously implemented to ensure that values were modified as desired and correct procedures were called.

By conducting single-agent testing, several errors came to light, as well as multiple possible improvements to ensure undesired behaviour could not emerge. This was done by attempting to break the model with extreme and unrealistic or impossible values, as well as conducting several

sanity checks for general model behaviour. It was found that the model implementation operates as desired for the most part, and errors encountered were immediately resolved.

This was further put to the test during interaction testing within a minimal model. This step entailed devising several scenarios for agent and system parameters. Each scenario was accompanied by a set of checks for parameter modifications that should occur and the order at which these occur. No errors were found during this stage, indicating that the structural validity of the model is likely in line with the concepts identified in chapter 2, chapter 3 and chapter 4.

The last step, multi-agent testing, was conducted to verify whether output values could logically be explained. It was identified that the nature of the model implies chaotic behaviour when assessed over the timeline of a simulation run. This is the direct result of the reactive nature of the simulation model, as there is no developing degree of model performance over time. For this simulation model, the main purpose of the timeline is to prescribe the amount of data points required for thorough analysis. It was found that within single model runs, the variability between subsequent time steps could be large, while the general tendencies of each model run were explained by occurring events, such as an active attack emerging. The variability between multiple model runs was found to even out across a multitude of repetitions, which was in line with expectations. The main take from this is that analysis relating to the time scale is not likely to yield any noteworthy results. Instead, results should be assessed as cumulative parameters relating to the entire model run. That way, performance of the model is stable across several model runs, which paves the way for more thorough exploration of experiments. Overall, the designed simulation model is in line with the desired formalised model and is capable of simulating interaction among entities and concepts identified previously. To ensure bias or sensitivity is not an issue, the experimental design should account for the desired model output parameters as well as possible sensitivity to uncertain model parameters.

## 6.6 Intermediate findings

This chapter sought to translate conceptual elements from chapter 5 into a fully-fledged simulation model. The first step to formalising the conceptual model was to identify which deviations would be applied to the conceptual model to avoid overcomplicating the formalised model. The main deviations made were switching the user agent group to a more passive role, as differences between different users were impossible to establish in a meaningful manner, and to synthesise defender agents and node objects, as defender agents could only account for behaviour related to a single node in the conceptual model to begin with. The next step was to describe the narrative for model operation, indicating which procedures would be executed and the order in which this occurs. The main step towards model implementation was to specify concepts in the model to implementation-ready computations and mechanisms. After these concepts were specified, the model was implemented in NetLogo. Lastly, the simulation model was verified to ensure the implemented model was in line with the desired model. This was a crucial step before the model can be used for experimentation, as it contributes to the validity of eventual results given the existing exploration of each concept throughout previous chapters. The next step is to use the model for experimentation and subsequent data analysis to assess model performance.

# 7 Model experimentation and exploration of sensitivity

This chapter details the process of using the verified simulation model to conduct experimentation and the first phase of data analysis on the experiment results. The first phase of data analysis entails exploration of model behaviour under different circumstances. The first step towards producing a set of representative experiments is to identify the framework used for experiments and the parameters used for experimentation. This is explained in section 7.1. The second step, discussed in section 7.2, describes the results of model exploration and the implications for further data analysis. The results from experimentation and exploration will be wrapped up in section 7.3, providing intermediate findings.

## 7.1 Experimental design

The aim of this section is to detail the ways in which the simulation model can and will be used to explore and assess system behaviour under certain parameter configuration. First, the framework applied to exploration and experimentation will be discussed. Secondly, model parameterisation is discussed. Thirdly, the setup for experimentation is described.

### 7.1.1 Exploratory modelling and analysis and *Evaludation*

The central question posed for the modelling objectives was previously defined as *"What happens when …?"* (Nikolic & Kasmire, 2013). The main goals for model usage are exploratory in nature, seeking to analyse deeply uncertain and complex system behaviour. Bankes (1993) notes that exploratory modelling can help address issues typically uncovered by traditional consolidative modelling, as exploratory modelling can provide insight into the effectiveness of policy configurations even when models are not validated and true statistical sensitivities are unknown. Since this study revolves around ecosystem-level interaction among tentatively represented agents, it is difficult to establish the true meaning of entities in the model. While attempts are made to validate the model to true real-world behaviour and outcomes, deviations in defensive strategies by definition stray into the unknown. Despite the degree of uncertainty, validity of exploratory models can be established by coherently integrating substantiated conceptual facets into eventual model outcomes (Augusiak et al., 2014). Given this degree of uncertainty, experiments should account for possible subsequent deviations in system behaviour.

In order to establish experiments that can achieve these objectives, the *Exploratory Modelling and Analysis* (EMA) approach for simulation models as described by Kwakkel and Pruyt (2013) will be applied. The EMA framework delineates a set of practices and checks for exploring model behaviour. This approach is tailored specifically to using simulation models to analyse technological interventions, highlighting their effectiveness across multiple performance indicators. Since the uncertainty experienced on an ecosystem level within this study cannot be reduced, the EMA approach helps achieve valid simulation results. Connecting to the notion of emergent behavioural patterns for complex adaptive systems, EMA tries to extract data regarding these patterns from a vast set of experiments with different parameter settings.

First, a set of uncertain parameters is established for the model. This includes interactions that are based on values that are likely to vary to a certain extent. Along with the set of parameters, their value ranged are also specified. The second step is to establish data analytics that cover the variety of system runs, identifying the degree of uncertainty experienced for certain performance metrics. The third and last step is to discuss the results of these analytics in the light of policy representations to indicate under which circumstances proposed interventions could be effective. Crucially for agent-based simulation models, the analytics can be used to point out the value ranges for which certain policy interventions are effective (Kwakkel & Pruyt, 2013). For the purposes of this study, EMA can be

used to assess robustness of conceptual defensive strategies across the possible scenario space. A framework that is typically applied to exploratory models is the *Evaludation* approach by Augusiak et al. (2014) briefly introduced in section 6.5. *Evaludation* helps establish valid results from exploratory models and will be used to validate model outcomes in chapter 8.

### 7.1.2 Experimental design

Based on the EMA framework, a design was generated for experimentation. The experimental design consists of a set of scenarios for model parameters and designs for defensive strategies. In this sense, an experiment is defined as the combination of a design and a scenario. The aim for experimentation is to explore the effects of formulated conceptual designs. The scenarios are included to explore the robustness of each design across of a variety of model parameter settings to account for model sensitivity. First, the setup for model parameters will be discussed, as well as the set of scenarios explored. Afterwards, the implications of the set of parameters are discussed, denoting the number of experiments to be conducted as well as the data collection process. These parameters correspond with the formalisation of model concepts in Appendix B.

Conducting thorough and accurate experiments requires formulation of several factors crucial to this process. As discussed above, the main elements are designs for defensive strategies and parameters for scenarios that describe a certain degree of uncertainty (Kwakkel & Pruyt, 2013). Chosen parameter settings are based on default parameter values listed and substantiated in Appendix F. In order to produce valid experimentation results, it is necessary to establish the veracity of each parameter (Augusiak et al., 2014).

Four defensive strategies were conceptualised to be applied across the set of experiments. These are listed and described in Table 7-1 below. The associated values used for design parameters used for experimentation are listed in Table 7-2 below. There are four defensive strategies. Between all defensive strategies, thresholds for alleviation and retention are shifted to account for overall differences in false positive and false negative rates. The inclusion of operability as a central modelling construct requires an appropriate interface for decision-making. In this case, perceived operability serves as the metric used for deciding which responsive mechanism is required. Since the concept is in itself conceptual, the threshold used is a simplification of a complex process that in the real-world depends on many additional variables. The values used for these design parameters account for differences in sensitivity and specificity for each defensive strategy. These values were established through a process of trial and error, where a brief run of baseline experiments was used to establish which set of thresholds performed best in a static environment for each strategy.

The first strategy represents fully anomaly-based intrusion prevention and detection, likely leading to a low false negative rate. On the other hand, the associated false positive rate is likely higher. The second strategy involves fully specification- or signature-based intrusion prevention and detection mechanisms. The functioning of this strategy opposes the first strategy, as a raised alarm almost always equates to an attack taking place, yet attacks themselves are less likely to be thwarted at an early stage. The third strategy is a hybrid between the first and second strategies, deploying anomaly-based intrusion prevention and specification - or signature-based intrusion detection mechanisms. Similarly, the fourth strategy combines signature-based intrusion prevention and anomaly-based intrusion detection systems. These designs seek to extract the best qualities from both strategies.

*Table 7-1: Defensive strategies and expectations*

| *Strategy* | *Description/expectation* |
|---|---|
| *(1) Fully anomaly-based intrusion detection & prevention* | This strategy involves one of three typical intrusion detection and prevention systems (IDPS) as described by Patel et al. (2013) and seeks to detect patterns that deviate from pre-specified behavioural rules (Mitchell & Chen, 2013). These systems tend to be sensitive to raising false alarms, but are adaptive enough to deal with the vast majority of attacks. Integrating such a strategy for coherent, ecosystem-level assessment of threats to individual systems is likely to prevent many attacks from entering the system and spreading across connected nodes. However, the system is likely to cause cascading failures by filtering out crucial user traffic. |
| *(2) Fully signature-based intrusion detection & prevention* | This strategy is based on another intrusion detection and prevention system discussed by Patel et al. (2013). Signature-based intrusion detection and prevention involves matching detected patterns with specific types of attacks or misuse. These systems work well at detecting a known attack if present, without the negative side effects from the first strategy. However, these mechanisms fall short in dealing with an adaptive and constantly evolving threat environment (Berthier et al., 2010). This strategy is expected to perform best at preventing unnecessary defensive decisions, but likely falls short at consistently dealing with active attacks in a timely manner. |
| *(3) Hybrid between anomaly-based intrusion prevention & signature-based intrusion detection* | This strategy is based on a third category of intrusion detection and prevention systems suggested by Patel et al. (2013). These systems combine elements found in other IDPS in an attempt to achieve the best of both worlds. In this case, the deployment of a hybrid system uses two separate methods distinctly for intrusion prevention and intrusion detection. By applying anomaly-based intrusion prevention, the presence of attacks within the ecosystem is minimised. Subsequently applying signature-based intrusion detection helps detect the small margin of attacks that was not detected, while avoiding unnecessary complication of false alarms. The expectation is that this strategy works well at adapting to shifting circumstances within the ecosystem, but might fail to address core issues arising from critical user traffic. The decision to alleviate cannot be stopped in the model, and it can take one mistake in assessment to cause chain disruptions in the network. |
| *(4) Hybrid between signature-based intrusion prevention & anomaly-based intrusion detection* | This strategy seeks to strike a similar balance between anomaly- and signature-based intrusion prevention and detection systems as Strategy 3. In that sense, Strategy 4 is the opposite counterpart to Strategy 3, as it makes use of signature-based intrusion prevention and anomaly-based intrusion detection. In essence, this strategy seeks to mitigate the impact of erroneously blocked user traffic while ensuring attacks that might have slipped through are detected quickly. The expectation for this defensive strategy is that it encompasses more negative aspects of both types of controls than other defensive strategies, as inoperability resulting from unprevented attacks and obfuscation of situational awareness due to false alarms are both risks. Other defensive strategies encounter only one of these problems, whereas Strategy 4 likely incurs both. |

*Table 7-2: Design parameters for each defensive strategy*

| *Parameters* | *Strategy 1* | *Strategy 2* | *Strategy 3* | *Strategy 4* |
|---|---|---|---|---|
| *Prevention sensitivity* | 0.95 | 0.80 | 0.95 | 0.80 |
| *Prevention specificity* | 0.80 | 0.95 | 0.80 | 0.95 |
| *Detection sensitivity* | 0.95 | 0.80 | 0.80 | 0.95 |
| *Detection specificity* | 0.80 | 0.95 | 0.95 | 0.80 |
| *Alleviation threshold* | 0.70 | 0.80 | 0.75 | 0.70 |
| *Retention threshold* | 0.30 | 0.20 | 0.25 | 0.20 |

In order to explore the robustness of these conceptual designs across the uncertain parameter space, a set of scenarios is formulated, the value ranges of which are shown in Table 7-3. These parameters are varied based on their default value as denoted in Appendix F, ensuring that uncertainties surrounding the parameters do not dictate model behaviour. The low degree of tangible information available for ecosystem-level analysis of critical infrastructures complicates the validity of experiments to be conducted.

*Table 7-3: Scenario parameter value ranges*

| *Parameter* | *Value range* |
|---|---|
| *Dependency weighting* | $\geq 0.3\ and\ \leq 0.7$ |
| *Attack frequency* | $\geq 0.03\ and\ \leq 0.07$ |
| *Attack powers* | Attack 1: $\geq 0.25\ and\ \leq 0.45$<br>Attack 2: $\geq 0.45\ and\ \leq 0.65$<br>Attack 3: $\geq 0.5\ and\ \leq 0.7$ |
| *Worm spread likelihood* | $\geq 0.1\ and\ \leq 0.4$ |
| *User traffic frequency* | $\geq 0.2\ and\ \leq 0.6$ |
| *User traffic criticality* | $\geq 0.2\ and\ \leq 0.6$ |
| *Alleviation duration* | $5, 6, 7$ |
| *Retention duration* | $2, 3, 4$ |

The set of scenarios is created within these parameter ranges following Latin Hypercube Sampling (LHS). Nikolic et al. (2013) denote the applicability of LHS to agent-based models, as they generate a set of parameter settings that approximates uniformity across the scenario space. To accomplish this, a set of 250 unique scenarios was created using the 'lhs' package in *R* by Carnell (2016). Since the aim of these experiments revolves around comparing the robustness of four designs, a deliberate choice was made to generate a single set of unique scenarios to be used across all four designs. Exploratory modelling, especially using models that prove difficult to validate, typically involves a large number of model runs to ensure results are generalisable for possible real-world representations (Bankes, Walker, & Kwakkel, 2013). By incorporating a set of 250 scenarios and four designs, a total of 1000 unique experiments will be conducted. To ensure the effects of chaotic model behaviour on results are reduced, multiple repetitions are typically conducted for agent-based models, so long as computational requirements allow for this (Nikolic et al., 2013). Each repetition repeats the same experiment, but variation will still be encountered due to chaos. There is no 'right' number of repetitions, but the general rule of thumb prescribes simulating as many repetitions and scenarios as possible. Since nearly all model runs using identical parameter settings were verified to yield similar distributions of data (detailed in full in Appendix E), there is not a significant degree of variability for

model outcomes across multiple repetitions. To meet computational requirements, each experiment is conducted 25 times, yielding a total of 25000 repetitions, each repetition spanning across 1825 ticks.

### 7.1.3 Experimentation output

The experimental design was implemented within the agent-based model and ran fully on two separate machines to ensure no data corruption took place. The total set of experiments, spanning 25000 repetitions of 1825 ticks, yields a total number of observations of over 45 million. This number of observations would result in a dataset that would prove difficult to process on both machines available for this study. To mitigate this, experiments for each defensive strategy design were run separately and the outputs were pruned to only contain necessary information. Since the chosen performance indicators for this model require nearly 30 separate parameters to be tracked at each time step, the resulting dataset size was substantial at over 8 GB. The size of this dataset complicates the set of analysis tools prepared to achieve desired insights in model performance across experiments. The dataset was pruned to only contain the required data points at the end of each repetition, resulting in 1 data point per repetition instead of the original 1825 data points per repetition. As a result, the number of data points was reduced by roughly 99.7%, from over 34 million to 25,000. Both machines resulted in comparable runtimes of around 2.5 hours for full model simulation (approximately 40 minutes per defensive strategy design), for an average run time of less than half a second. Following these steps, the dataset was prepared for data analysis to answer the research sub-questions.

## 7.2 Model exploration

The aim of this section is to detail behaviour emerging from the experiments. The process of exploring model behaviour entails two main elements. First, model performance over the entire set of experiments is assessed for each performance indicator. Next, the impact of variations in scenario parameters is discussed. Together, these steps help establish the sensitivity of the model towards uncertain parameters, which is crucial in ensuring model outputs are valid and reliable (Augusiak et al., 2014).

### 7.2.1 Model performance density

Exploring model behaviour is no straightforward task, as the underlying complexity of concepts incorporated into the simulation model might interact in a way that was not expected. This also underlines why model exploration is so crucial, as it helps understand how the system might perform in the real world. This subsection seeks to establish the Kernel Density Estimations of model output across the entire set of scenarios to assess emergent patterns. This is done for a multitude of model parameters, following the same structure as applied to multi-agent verification described in appendix E. For each parameter, the expected behaviour is hypothesised and subsequently assessed based on presented data. The general expected behaviour if no particular sensitivities are found is symmetrical behaviour across each experiment, with positive and negative swings causing similar deviations from baseline model behaviour, albeit in different directions. Each plot will be drawn from the complete set of experiments and will therefore likely contain variability that is attributable to the effects of different defensive strategy designs. These plots were generated using the *ggplot2* package in R, providing clear and customisable customisation of many different visualisations (Wickham & Chang, 2008).

While density estimations alone do not provide complete insight into the performance of individual runs, variability analysis conducted in appendix E.IV shows that deviations among repetitions belonging to each experiment are negligible. The mean and standard deviation associated with each

performance indicator are listed in Table 7-4. The variability of model runs is discussed in more detail in appendix E. While deviations can occur, the extent of these deviations average out to almost perfectly symmetrical behaviour. Deviations observed across kernel density estimations can therefore be attributed to different model configurations used for different experiments.

*Table 7-4: Means and standard deviations for all performance indicators across 1000 repetitions for variability testing*

| *Performance indicator* | *Mean* | *Standard deviation* |
|---|---|---|
| *Cumulative losses* | 5566.92 | 343.53 |
| *Current losses* | 0.12 | 0.0075 |
| *Number of normal nodes* | 10.79 | 0.42 |
| *Number of stressed nodes* | 14.03 | 0.40 |
| *Number of inoperable nodes* | 0.18 | 0.054 |
| *Impact assessment deviation* | -0.059 | 0.0011 |
| *Fraction of 'nothing' decisions* | 0.39 | 0.0073 |
| *Fraction of alleviation decisions* | 0.60 | 0.0064 |
| *Fraction of retention decisions* | 0.0072 | 0.0021 |
| *Fraction of correct decisions* | 0.48 | 0.013 |
| *Fraction of overestimated decisions* | 0.030 | 0.0015 |
| *Fraction of underestimated decisions* | 0.49 | 0.014 |
| *Number of active attacks* | 0.21 | 0.027 |
| *Attack duration* | 2.66 | 0.098 |

*Incurred losses*

The first model output parameter analysed is the extent to which losses are sustained throughout a model run. This parameter implies a degree of direct, tangible performance for a model repetition, as preventing losses is the primary objective for defensive strategies. This is measured both cumulatively over the course of an entire model run and the average degree of contemporary losses per node over an entire model run. While values should differ, the patterns for these two parameters should be identical, as the latter is the result of summation of average losses per node for an entire run. The density plots for the extent of total losses are shown in Figure 7-1 below.

As shown in Figure 7-1, the behaviour shown by both plots is visually identical. This is to be expected, as both parameters relate to the same model parameter, except the former describes summates contemporary losses from all nodes and the latter averages these out. Assessing the density of model averages would therefore yield identical distributions for different value ranges. The interesting behaviour observed from these parameters is the asymmetrical shape of the density graph. The upper bound of kurtosis is significantly less densely distributed than the lower bound. In this case, scenario parameters that are more beneficial to node operation result in an even decrease in losses, as the majority of nodes is not subject to any significant inoperability. The upper bound is disrupted by the differences in distribution for defensive strategies. The lower bound instead approaches linearity down towards 0. The implications of differences between defensive strategies will be further explored in chapter 8.

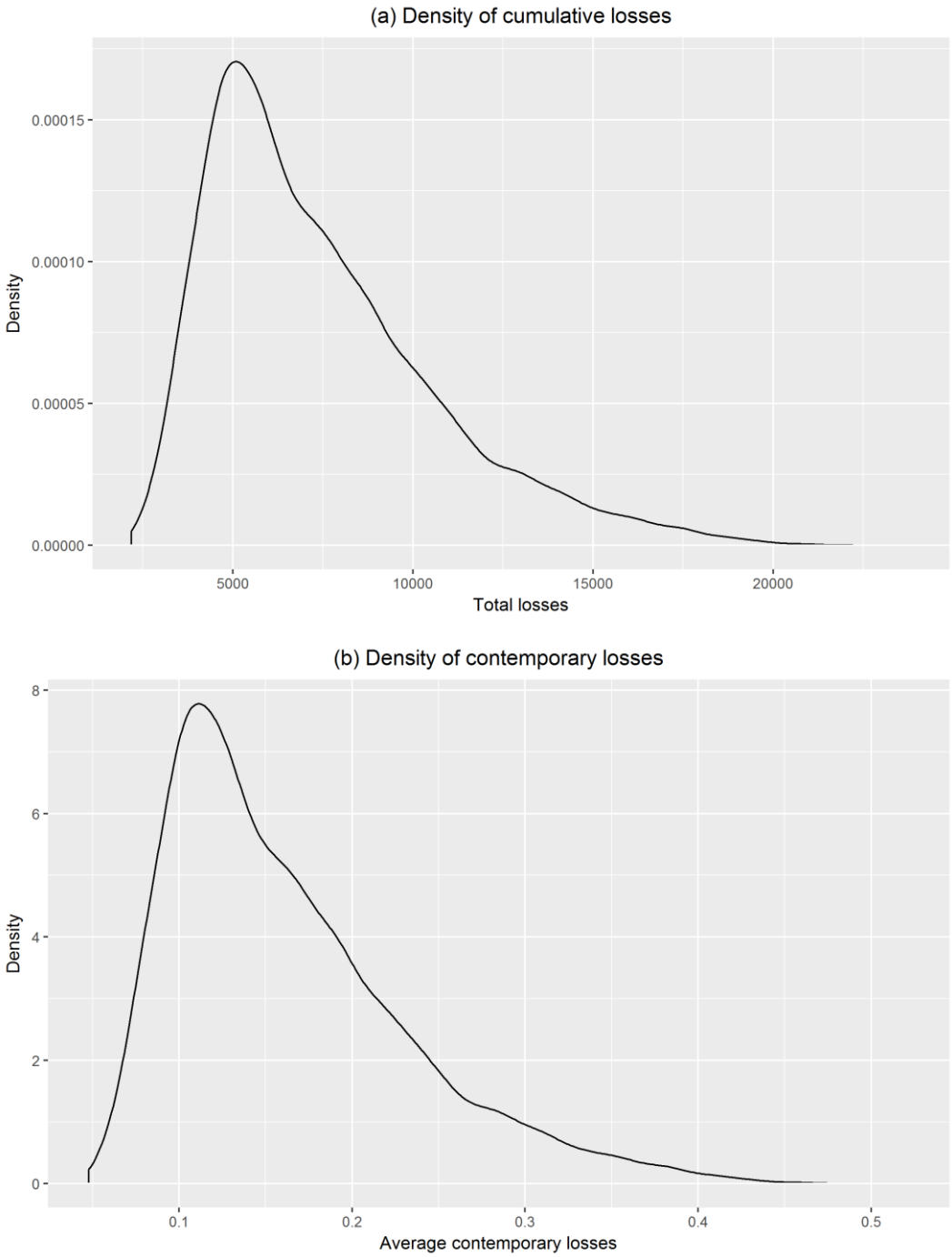


*Figure 7-1: Density plots for losses*

*Node operational states*

The second model output parameter analysed regards the operational states for nodes. Exploring the distribution of operational states for nodes helps understand the effects of defensive strategies and scenarios on how the aforementioned losses occurred. Similar loss values can be caused by either a large number of slightly hindered nodes or by a smaller number of inoperable nodes. The expectation for this parameter is that most runs are characterised by a relatively high number of nodes in normal and stressed operation, whereas the number of inoperable nodes is likely lower. This can be established based on variability testing conducted in appendix E.IV. Density plots for each node operation state are displayed in Figure 7-2 below.

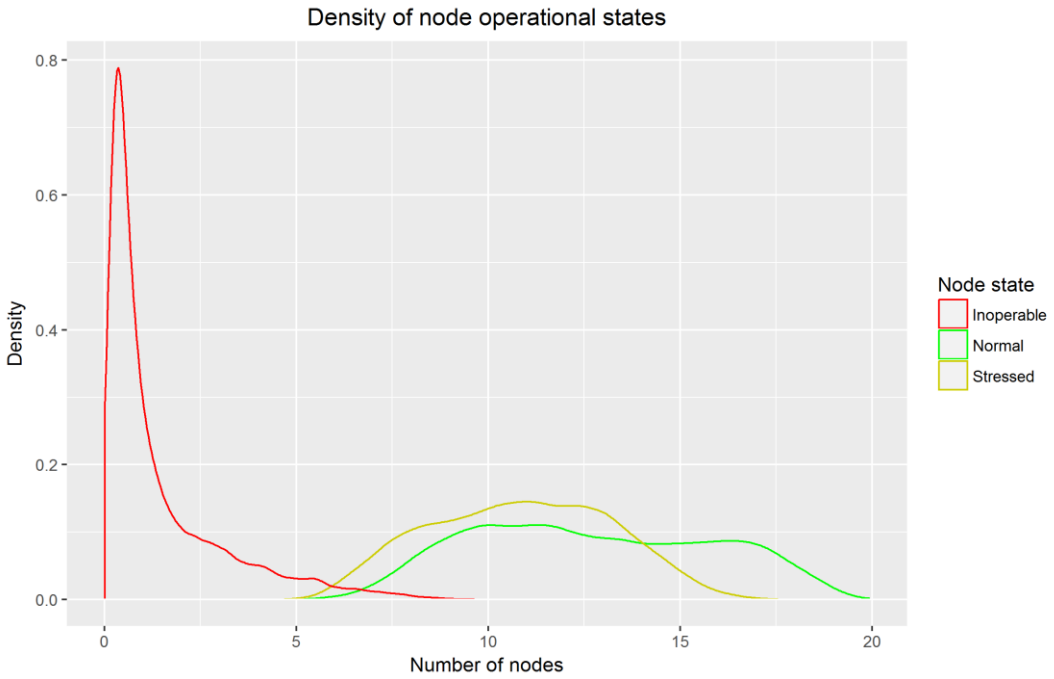


*Figure 7-2: Density plots for node and their operation states*

The density plots show general trends in node operational states. First of all, the majority of model runs show a marginal number of inoperable nodes on average. This is in accordance with expectations, as inoperability should be a rare occurrence in the ecosystem and be primarily attributable to key defensive decisions and cyberattacks. There are, however, several instances where model operation is disrupted to the point where over 5 nodes are inoperable on average. The impact this would have on model runs is significant. Noteworthy is that the shape of the density plot for inoperable nodes matches the disruptive growth in losses found in Figure 7-1, indicating that experiments with a high degree of losses are caused by multiple nodes being inoperable rather than slight disruptions in a larger number of nodes. The behaviour shown in terms of normal and stressed nodes is more natural and shows a certain equilibrium between clusters of model runs. A substantial portion of model runs have noteworthy numbers of stressed nodes, although there seem to be multiple local peaks that could be attributed to different defensive strategies. Relating these observations to single model runs observed in appendix E, node operational states are interchangeably stressed and normal throughout the majority of model runs.

*Average number of attacks*

The third model output parameter relates to the average number of attacks active at each time step across an entire model run. Keeping track of the active number of attacks during a run can help assess whether certain scenarios lead to a high number attacks, which in turn helps establish whether losses are attributable to internal or external impact components. On top of this, the fraction of attacks that are detected is also included, providing further information for this assessment. The density plots for the number of attacks and the undetected fraction of attacks are shown in Figure 7-3 below.

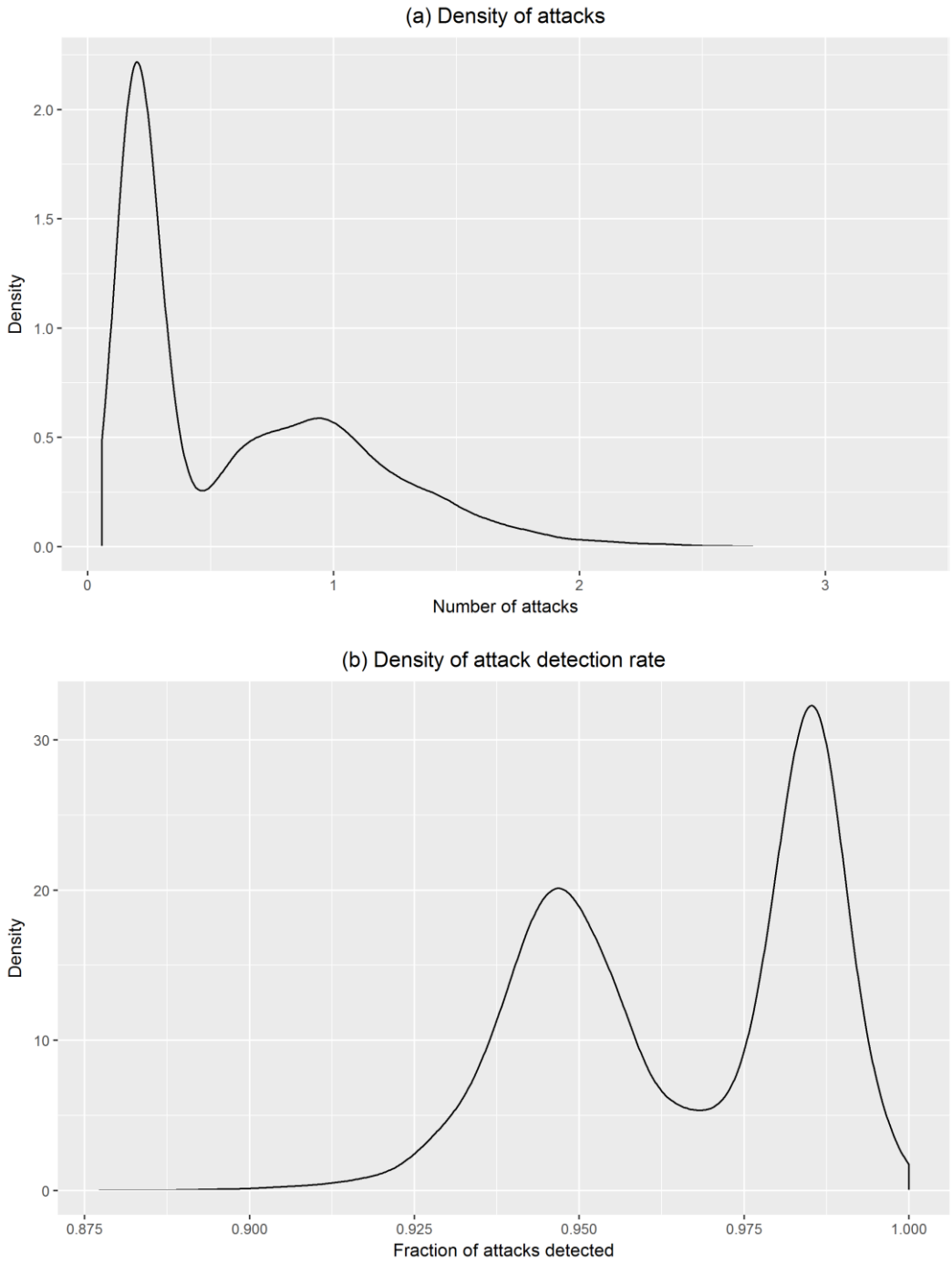


*Figure 7-3: Density plots for the number of attacks and the fraction of detected attacks*

The average number of attacks shows two distinct sets of patterns across the set of experiments. The most common results account for less than one attack per observation. This is to be expected, as this is limited by both the predetermined frequency at which attacks occur as well as the sensitivity of prevention mechanisms. Interestingly, another set of observations contains a higher number of attacks on average, forming a local optimum for a greater number of active attacks. The disparity across the set of results is far greater than variations in the frequency at which attacks occur. Therefore, it can be asserted that these variations are the result of certain defensive strategies failing to prevent attacks or prompt responsive measures in a timely manner.

This is corroborated by the similar shift in behaviour observed for the attack detection rate shown in Figure 7-3b, where a comparable fraction of observations shows a similar pattern. The observed deviations do not occur due to variations in scenario parameters, but instead follow implicit effects of defensive strategies. This will be discussed in further detail in chapter 8.

*Average duration of attacks*

The fourth output parameter involves the average duration of attacks. This indicates how long unprevented attacks are capable of inflicting damage upon nodes in the system before they are removed. The average duration of attacks is only accounted for if there are any attacks, as the metric would otherwise be convoluted by successful attack prevention. The duration of attacks is likely to vary significantly across a more widely distributed shape than previous parameters, as several scenario parameters directly affect capabilities to quickly remove attacks, and some defensive strategies might perform subpar at detecting attacks in a timely manner. The associated density plot is shown in Figure 7-4.

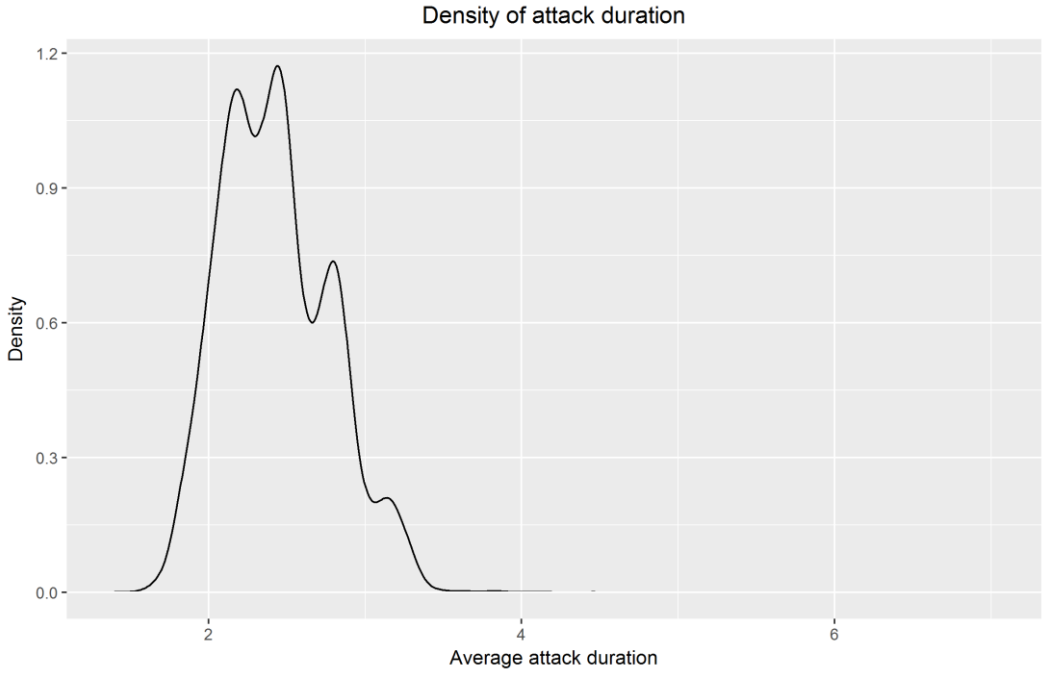


*Figure 7-4: Density plot for the average attack duration*

Behaviour shown in Figure 7-4 follows a largely symmetrical bell-shaped distribution, with four different local peaks. This implies that there is likely little variation based on specific value ranges for scenario parameters, as the general value range extends to upper and lower bounds of kurtosis. The more interesting observation is the overall range of observed values. In the most optimistic scenario for defenders, an attack would be detected instantly and alleviation duration and retention duration are respectively set to 4 and 2. In this instance, an attack would be removed after two time steps. Almost all values encountered indicate an extremely fast removal process, sometimes faster than the described optimal scenario. The only possible explanation for this is that single infrastructure nodes are repeatedly targeted by multiple attackers, while these nodes were already prompted to either alleviate or retain intrusions. That way, several newer attacks could be removed even if originally undetected or only briefly detected, as either responsive mechanism removes all inbound attacks. This brings down the average to values encountered in the density plot.

*Overall decision errors*

The fifth output parameter assessed is the fraction of response decisions made compared to the decisions that should have been made following a node's response thresholds and true level of operation. This helps establish to what extent defensive decisions are accurate, and more importantly, aids in drawing comparisons between decisions made following defensive strategies. Density plots were made of how frequent each type of decision (alleviation, retention, no response) occurred, as well as the correctness of decisions made across model runs. These are shown in Figure 7-5. Figure 7-5a depicts the former, whereas Figure 7-5b depicts the latter.

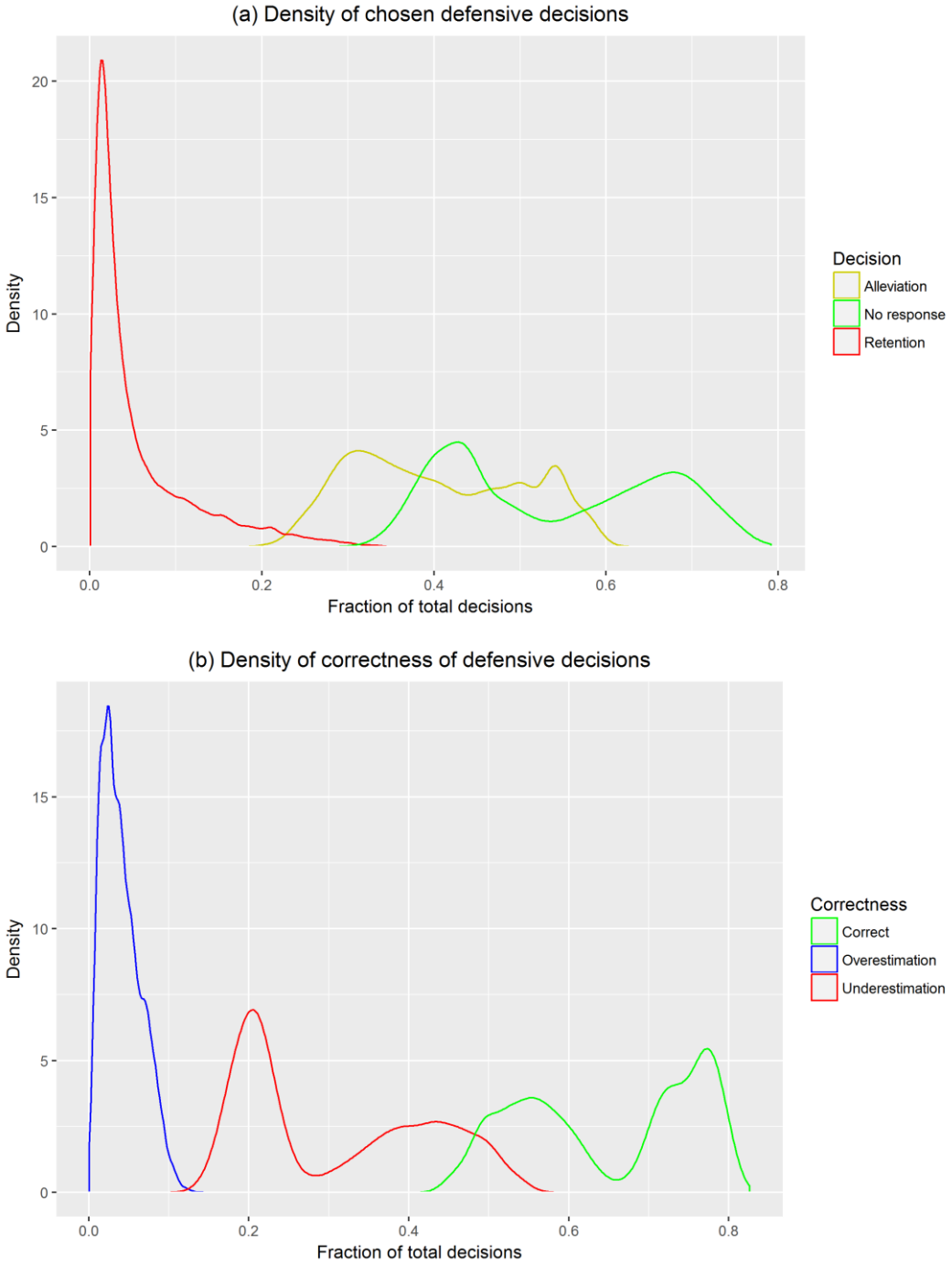


*Figure 7-5: Density plots for defensive decisions and their correctness*

Figure 7-5a indicates three main observations. The first, in line with expectations, shows that the majority of decisions in most runs is to not respond to intrusions and that retention decisions are incredibly uncommon. The reason for this is that responses are only warranted when an attack is currently active, and decisions are registered even when no attacks are active. Besides, the threshold for retention decisions typically requires multiple active attacks on one node, which is uncommon. The second observation relates to the oscillations in the upper bound of kurtosis for retention and alleviation decisions. A possible explanation for this is that separate defensive strategies react differently to certain combinations of parameter values. Through separating this graph in different facets for each defensive strategy, distinct oscillations were found in the behaviour for each defensive strategy, based on the (now smaller) variation in experiments iterated over. The combination of all defensive strategies therefore explains why certain value ranges seem to occur in troughs and peaks. The third observation relates to two separate clusters for the density of 'no response', as nearly all decisions are observed symmetrically in either of two clusters. This is likely related to observations in Figure 7-5b.

As seen in Figure 7-5b, behaviour shown across repetitions is stable for the number of decisions where the degree of operation was overestimated. Interestingly, the behaviour shown for correct decisions and decisions based on underestimating the degree of operability show almost identical yet mirrored behaviour. Different peaks and troughs in the density of values indicate distinct sets of experiments resulting in similar clusters for results. Because these variables are denoted as the fraction of overall decisions, an increase in one variable for one repetition is inherently tied to an overall decrease in the other two variables. Given the stability of 'overestimated' decisions, it is very likely that local peaks for correct decisions are associated with the similar, mirrored peaks in 'underestimated' decisions. The relative stability of experiments within these peaks indicate that either a small number of scenario parameters causes the distribution of observations into two distinct clusters, or that one defensive strategy leads to fewer correct decisions and more 'underestimated' decisions than the other two. In this case, the variance within clusters is attributed to difference scenario parameters, whereas the two separate clusters are caused by different defensive strategies. This will be further elaborated upon in chapter 8.

*Impact assessment error*

The sixth and last output parameter describes the average error made in impact assessment, given by the average deviation between perceived operability and true operability. Whereas the fifth output parameter denotes the fraction of decisions made erroneously, this parameter describes the extent by which impact assessment deviated from reality. Together, these parameters can help establish how well the concept of situational awareness was incorporated in a model run. The density plot for this metric is shown in Figure 7-6.

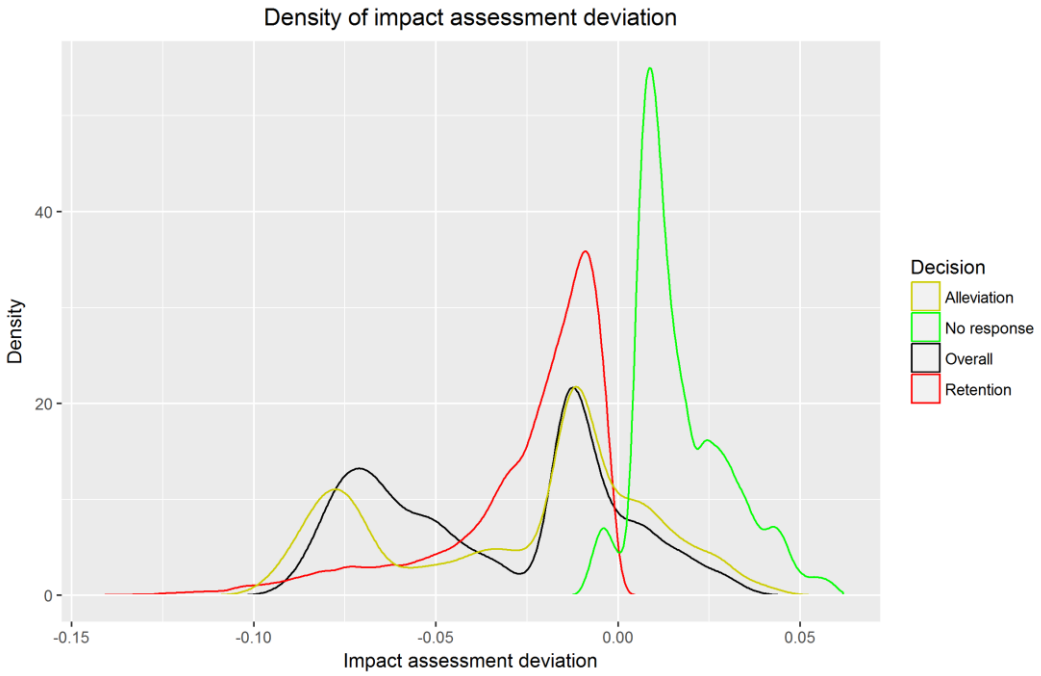


*Figure 7-6: Density plot for the average error in impact assessment*

The main observations drawn from the distribution of deviation in impact assessment relate to the differences in upper and lower bound behaviour. Under different system configurations, the deviation between impact assessment and true operability trends towards either positive or negative values. Positive values imply that, on average, impact assessment tends to overestimate the level of operation (either by not detecting true attacks or by erroneously preventing user traffic). Negative values represent the opposite, where the average impact assessment underestimates the level of operability, overstating the perceived effect of attacks.

The overall impact assessment deviation is in line with the behaviour identified for overall decision errors: the average trend results in a relatively low number of overestimating decisions, as the mean deviation is relatively stable around 0. However, two main clusters are identified: one asymmetrical cluster centred around -0.025 and another around -0.075. Similar clusters are found in the density of impact assessment for alleviation decisions, indicating a common root cause for this behaviour. The overall value range is rather narrow, as this metric involves the impact assessment of all nodes, including those not targeted by attacks. Since these plots involves the average error in impact assessment across entire model repetitions, such a stark difference and the visually distinctive manner it emerges highlight significant differences across multiple sets of model runs. Most of the overestimation can be explained by nodes opting to not respond or to alleviate, as these nodes have a clear alternative that should have been chosen (alleviation and retention, respectively). Further specification of the relationship between defensive strategies and deviations in impact assessment is provided in chapter 8.

### 7.2.2 Sensitivity towards specific parameters

Given the variations in system performance found in the previous subsection, the root causes for deviations in parameters are to be explored further. To this end, the data was arranged along subsets of parameter value ranges to explore whether any noticeable differences would emerge. Each parameter was measured against a subset of performance indicators used for initial exploration, with exception of the parameters *attack1-power, attack2-power* and *attack3-power*, which were grouped together to indicate the overall effects of higher and lower powers. If all performance indicators

were to be applied for exploration of parameter sensitivity, this would yield 64 graphs, each with multiple plots for value intervals. Most of these plots showed similar behaviour, which is why a selection was made as to what output parameters would be further explored. The selected output parameters are cumulative total losses, correctness of defensive decisions and the average error in impact assessment. The full results for this phase are listed and partially described in Appendix G. This subsection provides a summary of the most interesting fluctuations in system behaviour.

*Effects of dependency weighting*

The most direct and prominent effects witnessed across the set of scenarios relate to variations in dependency weightings. The overall impact of dependency weighting is supposed to primarily affect the total extent of losses inflicted to the ecosystem. Attack-induced inoperability in nodes translates over to dependent nodes, directly causing further inoperability. More heavily weighted dependencies further exacerbate the degree to which losses are incurred. These expectations are directly observed by sensitivity analysis, shown in Appendix G. The most variation is seen in Figure 7-7, which depicts the effects of dependency weighting on total losses incurred over time. The shape of the density plot shows significant shifts, as higher values for dependency weighting correlate with higher and less densely concentrated losses. The implementation of dependency weighting as a direct link between the operability levels is conceptually corroborated, but values used in the model are uncertain because of their ecosystem-level scoping (Setola & Theocharidou, 2016). Other performance indicators showed slight changes, indicating no significant sensitivity to extreme circumstances. Variations in dependency weightings are used to establish the robustness of defensive strategies, in line with the overall exploratory approach for data analysis.

*Effects of attack frequency*

Other interesting deviations were observed by conducting sensitivity analysis for different values of the frequency at which attacks occur. Variations in values for attack frequency directly affect the presence of threats to the ecosystem, as attacks become a more common occurrence. Since different defensive strategies deal with an increase in attacks differently, plots for different values of attack frequency should trend roughly similarly as a whole, yet show greater individual variation. This is observed most strongly in Figure 7-8, where the density of upper and lower distributions across single sets of parameter values show chaotic behaviour. Differences in the overall distribution between each group of parameter values are chaotic and the overall interpretability of these variations are largely negligible. This is likely attributable to different defensive strategy configurations handling several cases poorly, more so than with other scenario parameters.

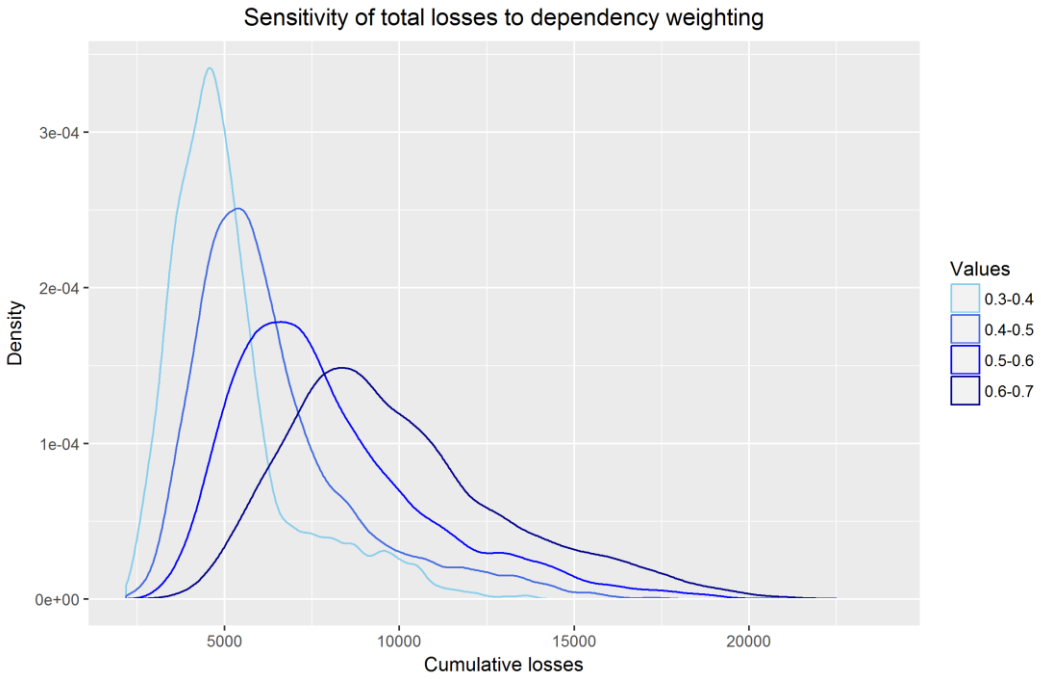


*Figure 7-7: Effects of dependency weighting on total losses*

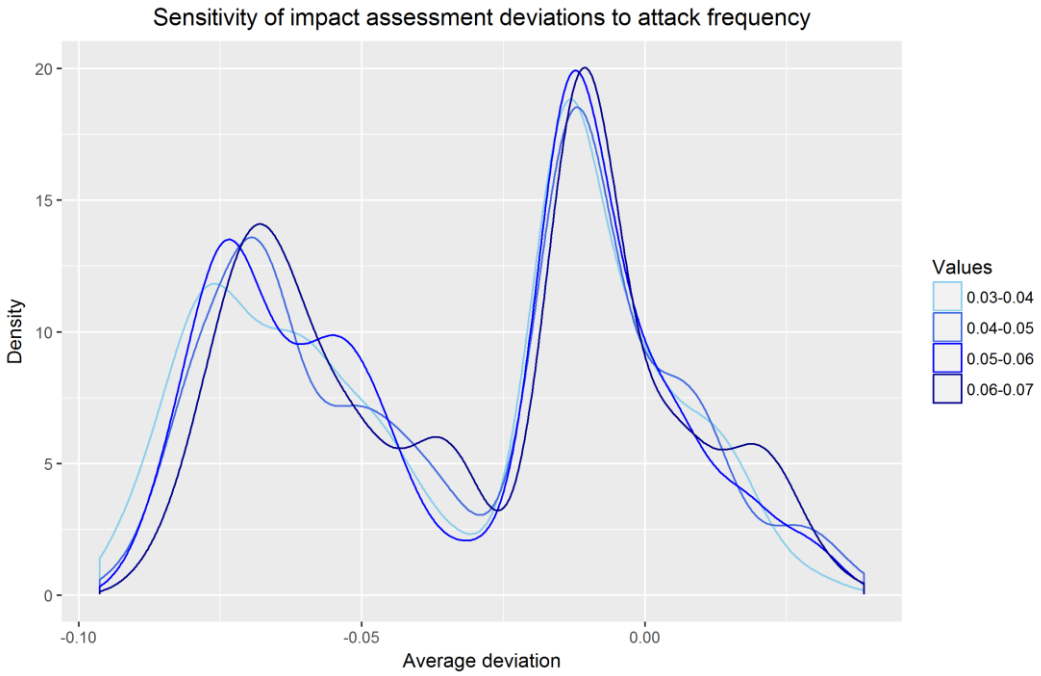


*Figure 7-8: Effects of attack frequency on impact assessment deviation*

*Effects of attack powers*

Values used for attack powers, as with the previous two scenario parameters, directly affect interaction that takes place between attackers and defenders. Attack powers are used for directly disrupting node operation, as well as obfuscating defenders' situational awareness through false positives.

If a false positive is generated during intrusion detection, defenders decrease their perceived internal operability by the power associated with the type of attack classified. Higher values for attack powers therefore affect nodes both directly through causing reduced operability and indirectly through

prompting incorrect decisions. The overall deviations in values for attack powers are relatively low, as their impact is based on historical attacks that disrupted parts of critical infrastructure systems (Miller & Rowe, 2012). For this reason, most variation across the set of experiments was relatively minor when compared to other scenario parameters. The most interesting observations relate to the average deviation between impact assessment and true impact, the density plot of which is shown in Figure 7-9.

The expectation would be that lower attack powers lead to generally more accurate impact assessment, since the positive deviation between perceived operability and true operability is lower when an attack is initiated and the negative deviation is lower for false positives. This pattern is most notably identified in the lower cluster and is more suppressed in the cluster of higher average deviation in impact assessment.

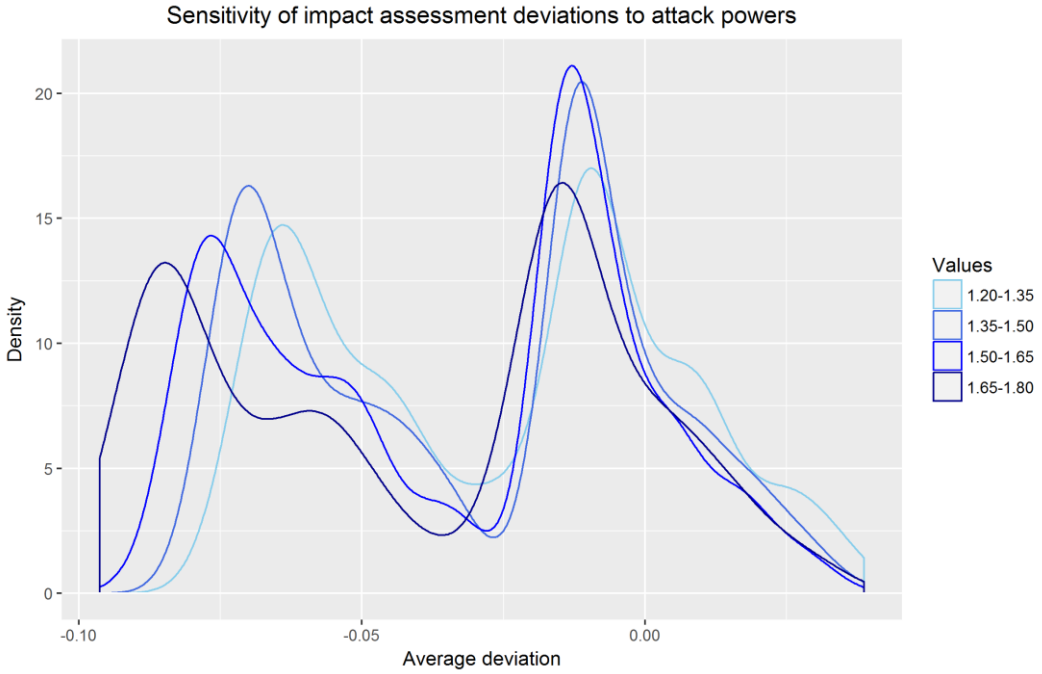


*Figure 7-9: Effects of attack powers on impact assessment deviation*

### 7.2.3 Implications for comparison of defensive strategies

The main findings discussed in this section relate to the sensitivity of key performance indicators to deviations in uncertain scenario parameters. Analysing a set of conceptual defensive strategies for a conceptual ecosystem involves entering academic uncharted territory. By extent, the process of modelling and experimentation depends on a great degree of uncertainty surrounding parameterisation (Bankes, 1993). By mapping the degree of uncertainty surrounding several scenario parameters, these variables can be varied across for simulation purposes. These scenarios were devised following the EMA framework and help establish patterns through thorough exploration of designated patterns in density plots (Kwakkel & Pruyt, 2013).

The main findings from conducting sensitivity analysis showed that most expected emergent patterns could be observed from kernel density plots for the total set of experiments. Plotting differences between sets of experiments with different value ranges for separate scenario parameters showed minor variations across most parameters. For each of these patterns, logical explanations could be formalised as to how fluctuations arise. There were a few instances of 'perfect storms', where some combinations of extreme parameter values would lead to crises, but none of

these crises varied too heavily from the mean model behaviour. The extent of variance discovered among the set of scenarios provides a solid basis for comparison of defensive strategy designs. Since there is not a single set of certain system parameters, these designs should be assessed in terms of robustness (Augusiak et al., 2014; Kwakkel & Pruyt, 2013).

## 7.3 Intermediate findings

Throughout this chapter, an effort was made to design a set of experiments that could be used to explore system behaviour under a multitude of different circumstances. To this end, the Exploratory Modelling & Analysis (EMA) framework was applied, as it helps establish valid results for models operating under deep uncertainty. Section 7.1 detailed how EMA concepts were integrated into the experimental design, as well as the parameterisation for each experiment. Section 7.2 detailed thorough exploration of model behaviour following the complete set of experiments. This resulted in insight into the robustness of system performance, which showed overall stability across variations of scenario parameters, with variations being explainable by model formalisms. These experiments will be used to conduct data analysis to assess the robustness of defensive strategies in the next chapter.

# 8 Data analysis and model validation

This chapter continues from where chapter 7 left off: analysing robustness of defensive strategy designs across the set of experiments. The main aim for this chapter is to produce all required insights to formulate answers to both sub-question 4 and 5. Section 8.1 details the process of analysing the differences between each design. These findings are validated in section 8.2 as part of a thorough assessment of model validity. The results from this chapter are synthesised and wrapped up in section 8.3.

## 8.1 Behaviour of defensive strategies

The aim of this section is to boil down the set of experiments to a set of favourable and unfavourable behaviours identified among the set of defensive strategies. Whereas chapter 8 aimed at identifying sensitivity across the set of experiments through exploration, this section seeks to compare the robustness of defensive strategies in order to formulate insights required to answer the main research question. Understanding how defensive strategies influence the ecosystem requires analysing how patterns for key performance indicators differ between strategies. This can eventually help establish which elements of defensive strategies provide desired effects.

The performance for strategies will be compared for each performance indicator, similar to how overall model performance was explored in section 7.2.1. The relative performance of each strategy will be discussed in the light of model formalisms and what the possible implication is for the effectiveness of a defensive strategy. This will subsequently be summarised into a coherent description of the implications of each defensive strategy.

### 8.1.1 Comparative performance of defensive strategies

This subsection will discuss the behaviour shown by each defensive strategy. Descriptions of defensive strategies from section 7.1.2 and their expected performance was previously detailed in Table 7-1.

*Overall damage and operation*

The first performance metric assessed is the robustness of each defensive strategy in terms of total losses sustained per simulation. To visualise this, a density plot was generated for each defensive strategy separately. This combined density plot is shown in Figure 8-1. The plot shows significantly different behaviour between the set of defensive strategies. Thinner shaped curves indicate more robust behaviour for a strategy. One strategy that stands out is the first, as it leads to substantially higher losses incurred over time compared to other strategies. Strategies 2, 3 and 4 show roughly similar patterns, specifically in comparison with strategy 1. Overall, strategy 2 performs best, as the vast majority of observations result in lower losses than strategies 3 and 4. Out of these similar strategies, strategy 4 performs somewhat worse, consistently resulting in slightly higher losses. The implementation of anomaly-based intrusion detection, unique to strategies 1 and 4, possibly affects the extent of node operability to an exacerbated degree.

Figure 8-2 depicts how these losses are sustained by visualising the density of operational states for nodes under each defensive strategy. Several striking observations are made. First, Strategy 1 shows a substantial cluster of inoperable nodes, while inoperable nodes are almost never observed for the other strategies. The distinctively higher losses for strategy 1 observed in Figure 8-1 are also observed here. Secondly, strategies 2 and 3 show nearly identical behaviour with relatively high density of nodes in normal operation, a lower density of stressed nodes and virtually no inoperable nodes. Furthermore, strategy 4 shows similar behaviour, although for this strategy stressed nodes

are more common than nodes in normal operation. The last observation underlines the slight increase in losses for this strategy identified previously.

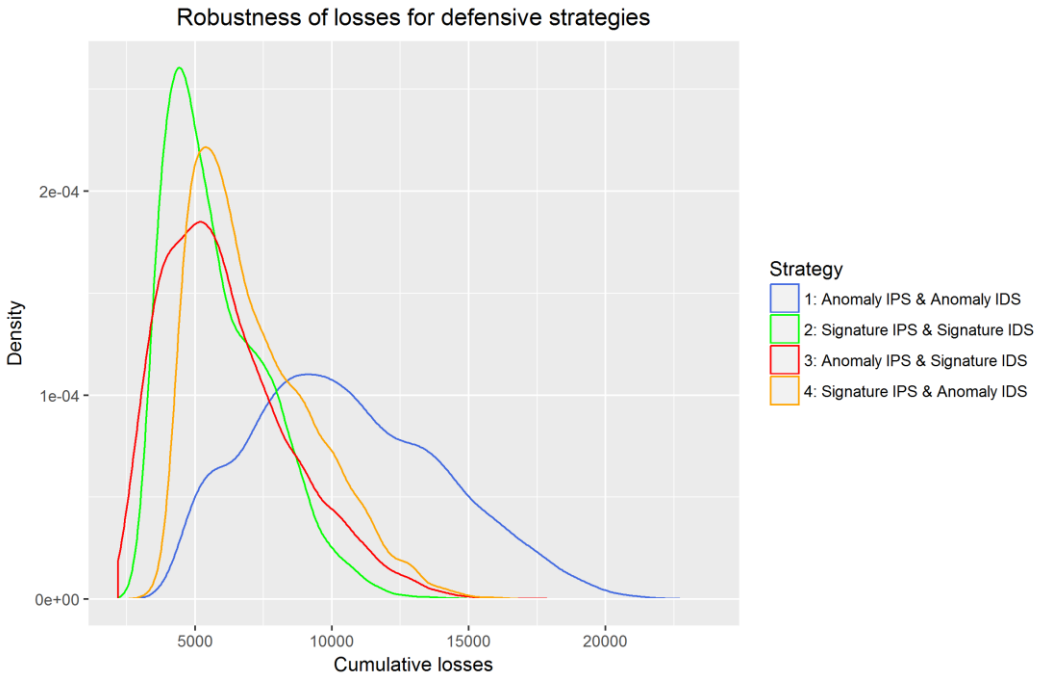


*Figure 8-1: Robustness of defensive strategies for cumulative losses*

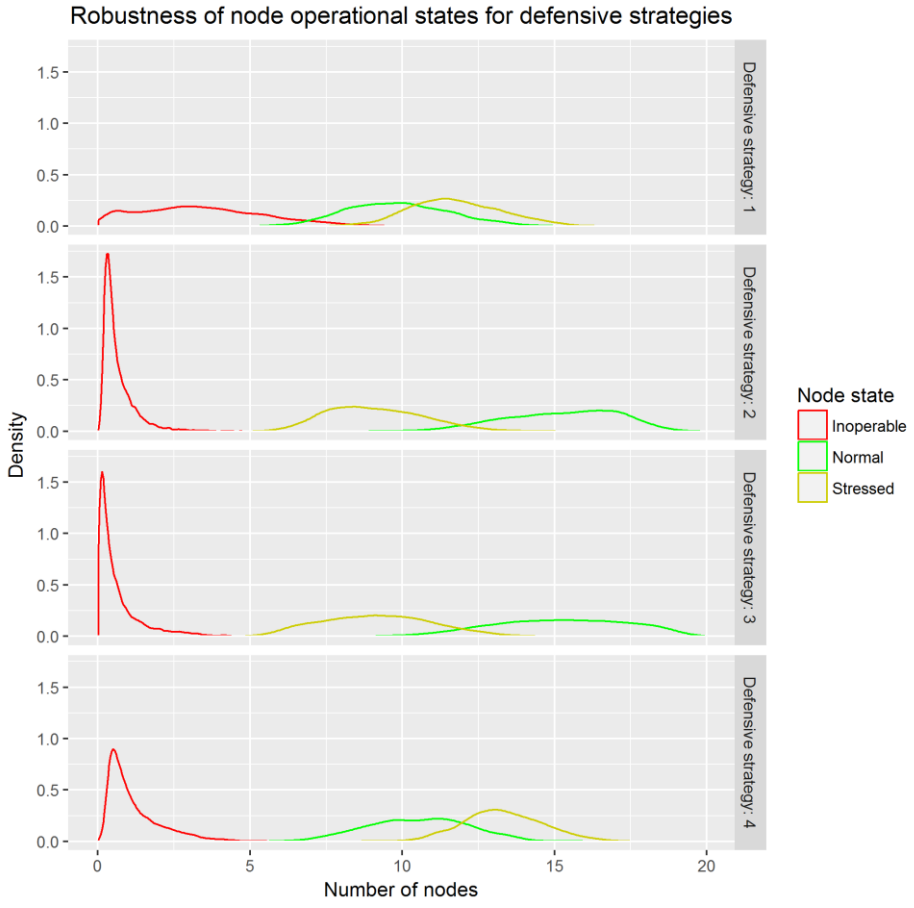


*Figure 8-2: Robustness of defensive strategies for node operational states*

*Cyberattack effectiveness*

The next step in assessing the robustness of each defensive strategy is to explore metrics related to cyberattack effectiveness. The average number of attacks active in the network is shown in Figure 8-3, again in the form of a density plot. Two striking patterns can be identified: strategies 1 and 3 are characterised by narrow density graphs encapsulating typically low values indicating few attacks were incurred, and on the other hand strategies 2 and 4 both show similar and less robust distributions around higher values. The latter implies that under strategies 2 and 4, successful cyberattacks were common events. The interesting take from this is that strategy 2 showed the most robust performance in terms of sustained losses previously, yet is also the least effective at preventing cyberattacks. This directly implies that under strategy 2, nearly all losses sustained are attributed to a select few nodes, while other nodes barely experience inoperability.

The higher frequency of successful attacks for strategies 2 and 4 is expected, since these are the only strategies that incorporate signature-based intrusion prevention, which is less effective at preventing attacks. However, the fact that strategy 2 showed the most robust density for losses incurred in spite of the high number of active attacks suggests that the majority of losses depicted in Figure 8-1 are attributable to defensive decisions made and are largely the result of unwarranted deployment of responsive mechanisms. The suggestion that false alarms pose the comparatively larger threat within this ecosystem will be further put to the test later on in this section.

Besides the number of active attacks, further suggestions about the relative performance of strategies can be deduced from the attack detection rate of and the average attack duration recorded. These metrics are shown in Figure 8-4 and Figure 8-5, respectively. The attack detection rate shows strategies 1 and 4 perform well at detecting any present attacks, as nearly all attacks are detected. Both of these strategies are characterised by anomaly-based intrusion detection systems, catching out all but a few attacks. Interestingly, strategy 4 results in a thin and symmetrical distribution, showing greater robustness than strategy 1. The same pattern is observed for strategy 2 in comparison to strategy 3. Both strategy 2 and 4 make use of signature-based intrusion prevention, suggesting that signature-based intrusion prevention systems cause attack detection to be more robust. This is likely the result of a higher number of unprevented attacks to detect, leading to individual simulation observations to be statistically less affected by single events. Analysis of average attack duration across defensive strategies fails to add any interesting insights, as it merely shows that strategies 2 and 3 take longer to respond to active attacks, which is logically the result from a lower attack detection rate.

However, merely analysing these metrics in isolation fails to tell the full story on the robustness of defensive strategies. It is clear that anomaly-based intrusion detection effectively reduces the strain experienced from cyberattacks while subjecting the ecosystem to an exaggerated perception of the threat landscape.

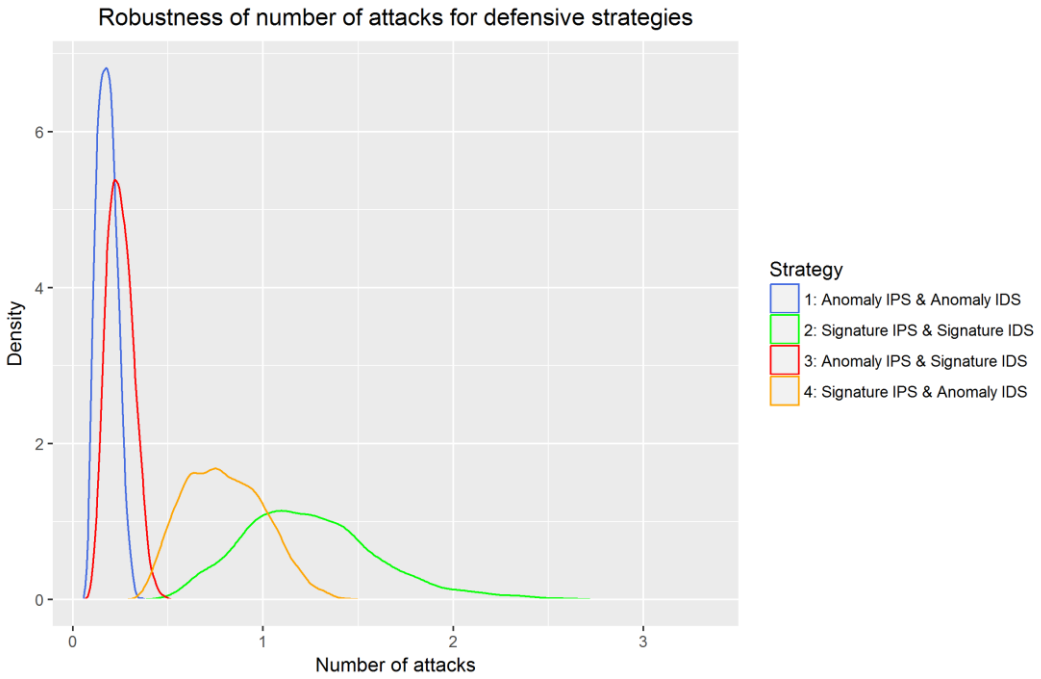


*Figure 8-3: Robustness of defensive strategies for the number of active attacks*

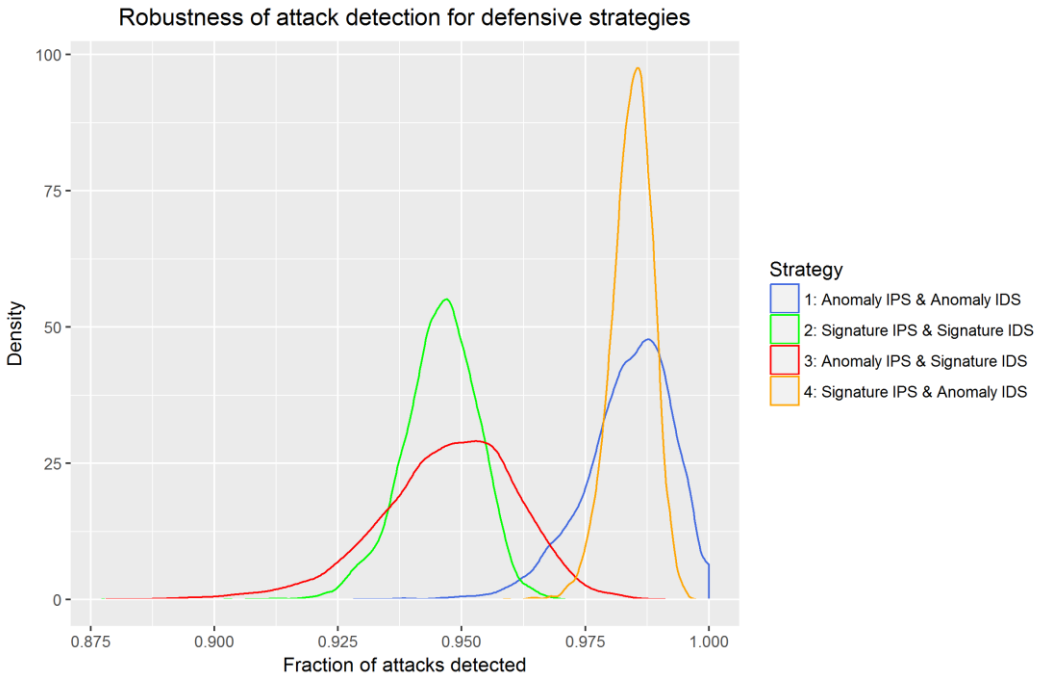


*Figure 8-4: Robustness of defensive strategies for the fraction of attacks that is detected*

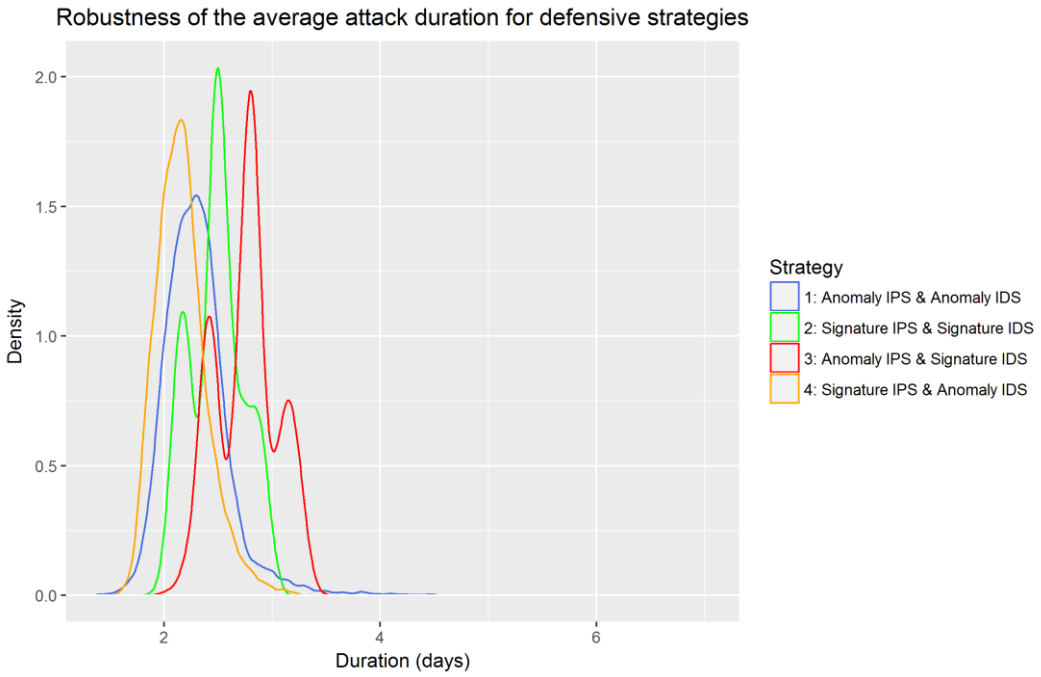


*Figure 8-5: Robustness of defensive strategies for the average attack duration*

*Frequency of defensive decisions*

The third step in assessing the relative performance and robustness of defensive strategies is to assess the density of each type of decision made. This adds another piece to the proverbial puzzle by providing an overview of how previously discussed metrics contribute to possibly beneficial or impeding defensive decisions. Figure 8-6 depicts the density plots of the frequency of each type of decision for each strategy.

Figure 8-6a and Figure 8-6b show the direct effects from the hypothesised exaggeration of the threat landscape by defenders for strategies 1 and 4, as the number of alleviation decisions is substantially higher. This is particularly apparent for strategy 4, as the majority of decisions made in the network result in alleviation. Both alternative strategies, being 2 and 3, seem to result in more passive decision-making, highlighted by the more common observation of decisions to not respond. Similar observations apply to Figure 8-6c, with the key difference being strategy 1 leading to a substantially different distribution of decisions. Out of all four strategies, only strategy 1 results in common retention of intrusions. This is not attributable to the configuration of its control mechanisms, as this would share some similarity with either strategy 3 or strategy 4. Instead, this is the result of a slightly higher value for operability threshold to retain intrusions resulting in a higher tendency to retain intrusions.

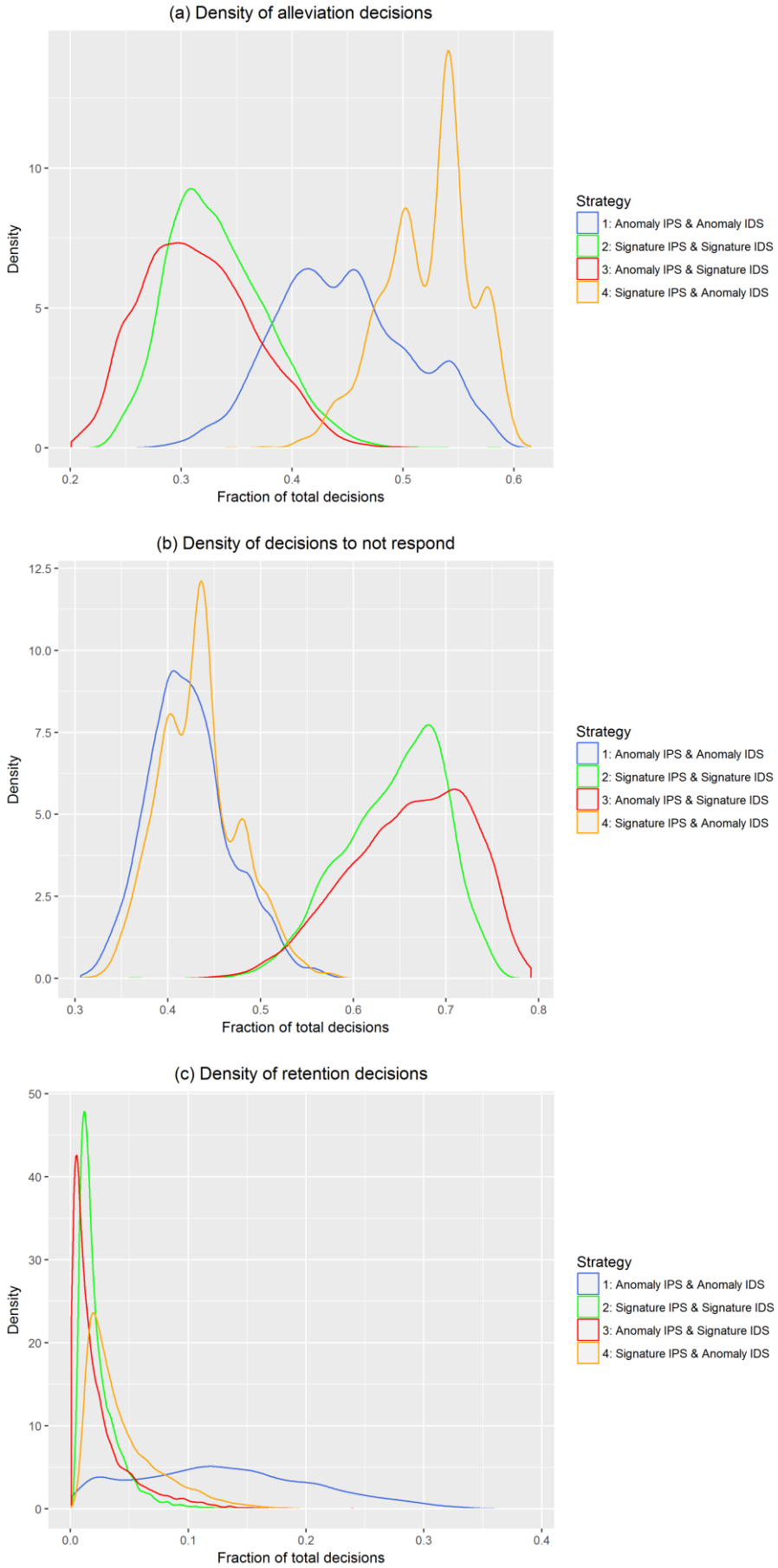


*Figure 8-6: Robustness of defensive strategies for the distribution of defensive decisions*

*Correctness of defensive decisions*

The last metric used for comparison between defensive strategies is the correctness of previously specified defensive decisions. All previous measures together can help hypothesise the implications for observations, but fail to show the number of correct and incorrect decisions or the average error in impact assessment. The density of the correctness of decisions is shown in Figure 8-7. Incorrect decisions are either based on overestimating node operability, for example through failing to detect an active attack, or underestimating node operability, for example through raising a false alarm.

The three plots together tell a clear story about the implications of the usage of different control mechanisms in terms of robustness and overall performance for defensive strategies. Based on Figure 8-7a, Strategy 2 results in the highest number of correct decisions made across the set of experiments. Strategy 3 follows suit with a similar yet slightly lower distribution of correct decisions. Strategies 1 and 4 lead to substantially lower distributions. The common factors between these pairs are strategies 2 and 3 sharing signature-based intrusion detection systems and strategies 1 and 4 both sharing anomaly-based intrusion detection systems. Signature-based detection systems are logically going to lead to fewer incorrect decisions, as the prespecified specificity determines. Analysing Figure 8-7b, it becomes clear that strategies 1 and 3 share similar tendencies to make more decisions based on an overestimation of operability than strategies 2 and 4. Strategies 1 and 3 both make use of anomaly-based intrusion prevention mechanisms, which result in a higher false positive rate than the signature-based intrusion prevention mechanisms used by strategies 2 and 4. Within the model, false positives during intrusion prevention lower node operability while situational awareness is unchanged. Therefore decisions following these events are more likely to be based on overestimated operability, as defenders are unaware of the impact. Figure 8-7c recounts the observations discussed for Figure 8-7a, as strategies 1 and 4 result in a substantial fraction of decisions made based on underestimating operability. Their anomaly-based intrusion detection systems raise comparatively unmanageable numbers of false alarms. Conversely, strategies 2 and 3 result in a lower fraction of decisions based on underestimation. The slight deviation between these two strategies found in Figure 8-7a is the result of strategy 3 making additional errors in judgment due to erroneously prevented user traffic.

The average deviation in impact assessment is shown in Figure 8-8a, which matches the descriptions provided above. Strategy 2 provides the most robust and balanced performance in terms of impact assessment, as this strategy avoids the problems discussed for both anomaly-based intrusion prevention and anomaly-based intrusion detection. Strategy 3 results in a tendency to overestimate operability, as was observed in Figure 8-7b. Strategies 1 and 4 more commonly underestimate operability than not. By plotting the origin of deviations in impact assessment, as presented in Figure 8-8b, additional insights are gathered. Strategies 1 and 3 commonly overestimate operability when no response was decided on, although the former results in multiple instances where alleviation and retention decisions are made on severe underestimations of operability and the latter shows more robust behaviour across its decisions. Strategy 2 on the other hand shows that the robustness in impact assessment is traceable back to all three types of decisions, and it seems that a delicate balance is struck. Strategy 4 diverges from the other strategies with a consistent tendency to underestimate operability regardless, although overestimations are only a rare occurrence, as with strategy 2.

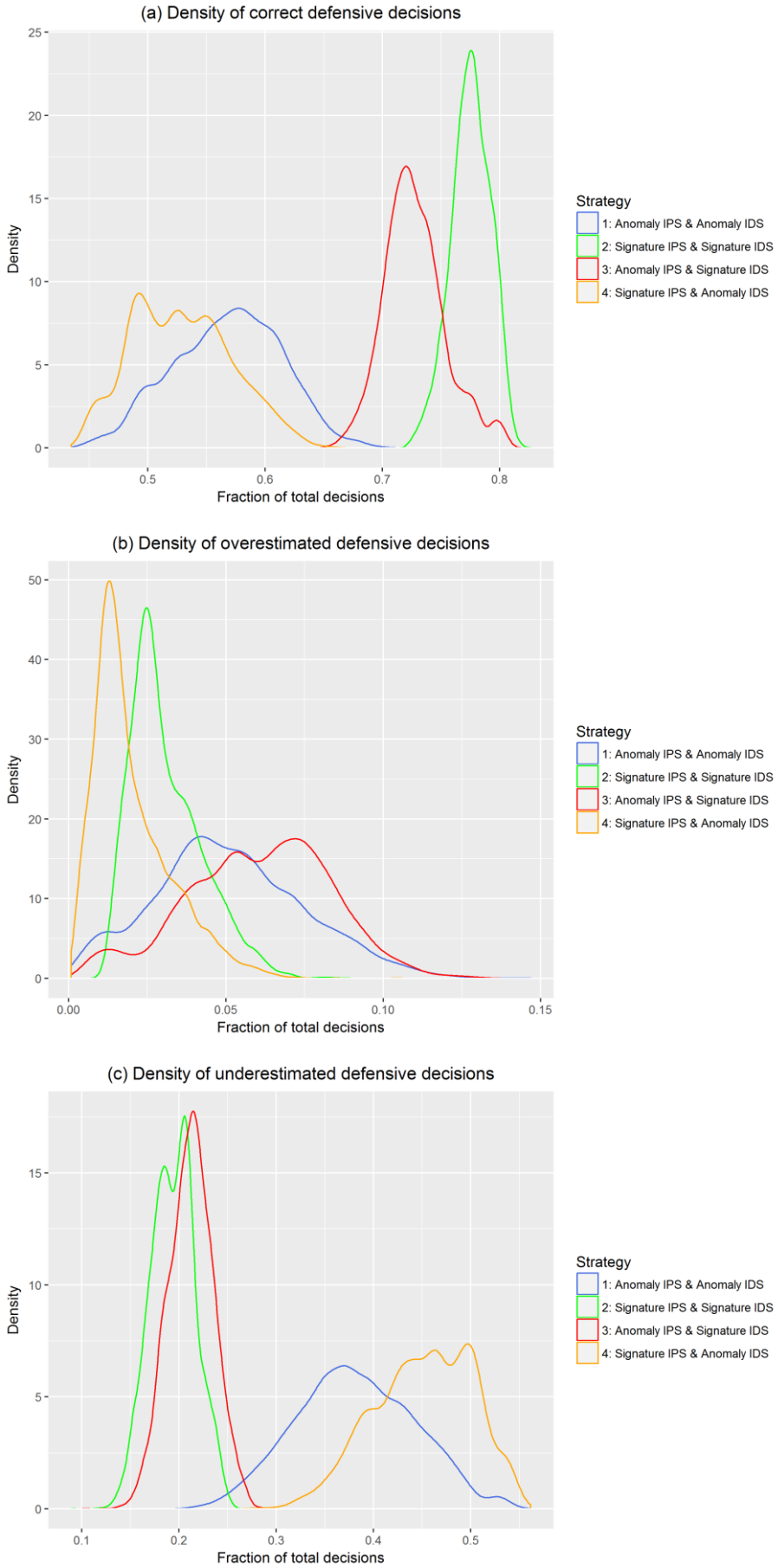


*Figure 8-7: Robustness of defensive strategies for the correctness of defensive decisions*

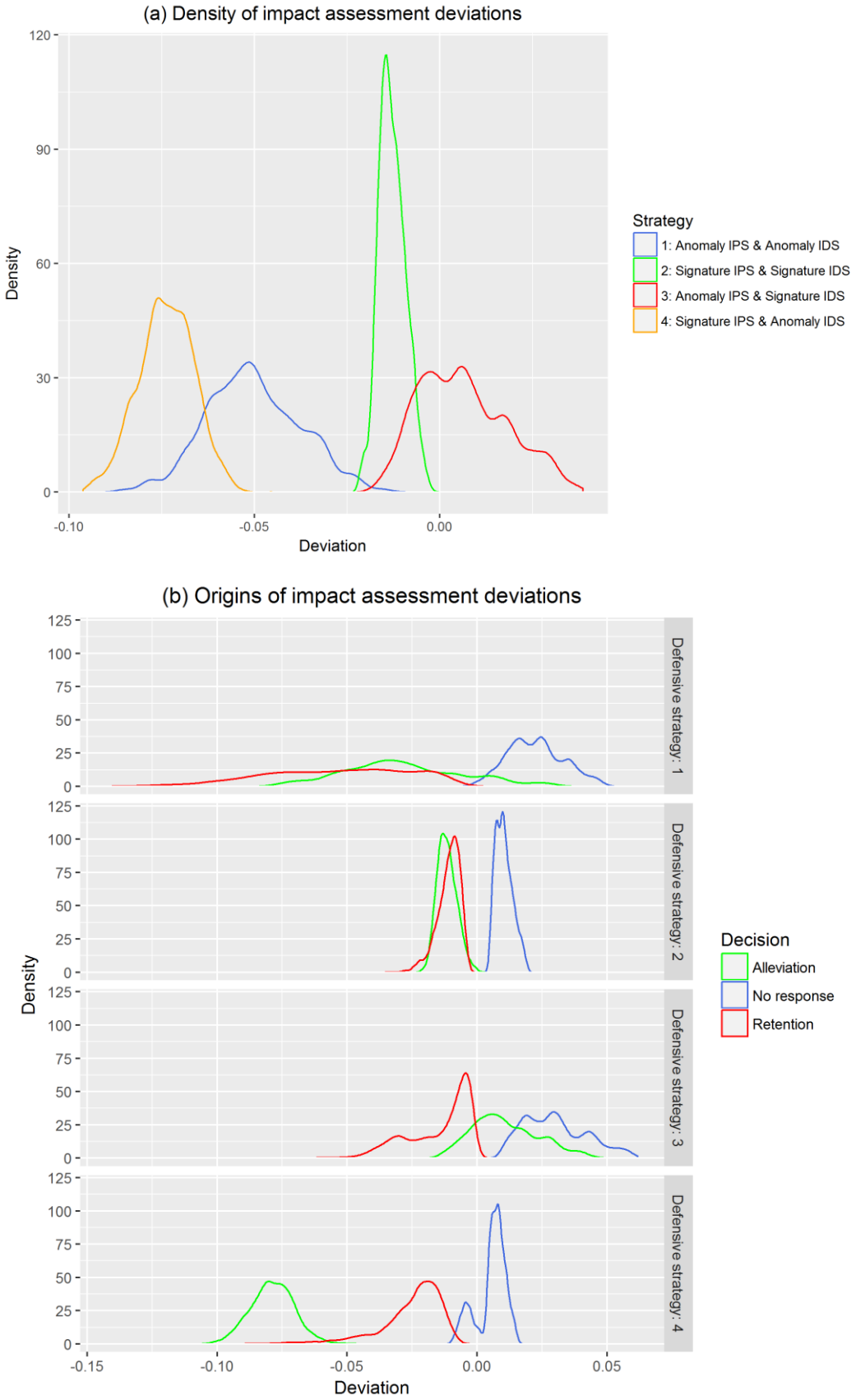


*Figure 8-8: Robustness of defensive strategies for the deviation in impact assessment*

### 8.1.2 Summary of defensive strategies

After exploring model behaviour for sensitivities to scenario parameters and trying to draw a comparison between defensive strategies, an increased understanding of how the ecosystem model responds to different scenarios and defensive strategies has been established. The main behavioural elements and the relative performance of each conceptual defensive strategy are summarised in Table 8-1.

Crucially, strategies 2 and 3 show comparatively robust behaviour, with one major difference. While both strategies result in a relatively low degree of losses incurred across model runs, strategy 2 is associated with a major caveat, as attacks are significantly more successful in terms of detection rate and average duration. The same caveat applies to strategy 4, although this strategy performs worse in other categories. Defensive strategies that incorporate signature-based intrusion detection result in consistently accurate decision-making, while strategies using anomaly-based intrusion detection result in obfuscation of situational awareness. On the other hand, defensive strategies that use signature-based intrusion prevention cannot mitigate and thwart cyberattacks as well as other strategies. While the effects of this modelling assumption seem logical in theory, the direct interpretability of results is therefore subject to thorough discussion. While the frequency of cyberattacks on critical infrastructures has been steadily increasing, successful attacks are still a rare occurrence. Within the ecosystem model, these attacks are a rare occurrence – the maximum possible losses from not dealing with attacks are therefore rather marginal and an equilibrium can be reached between consequences from false negatives and consequences from false positives. This works well under the assumptions this simulation mode is based on, but real-world examples of attacks have showcased the insurmountability of attacks. Generalisations and assumptions that were necessary to cover most ecosystem-level concepts might not hold up well when compared to real-world incidents, even if such modelling constructs were based on a solid foundation of academic literature.

## 8.2 Model validation

Model validation is a crucial step in the agent-based modelling cycle, as it establishes whether the modeller managed to “build the right thing” (Nikolic et al., 2013). Validating a model involves assessing whether the outcomes of a model correspond with observed patterns in the real world. Several methods exist for validating agent-based models, including historic replay of scenarios or expert validation of certain concepts. However, these methods do not account for problems encountered for exploratory modelling, as these models are often difficult to validate (Bankes, 1993). This leaves exploratory models in a juxtaposition between modelling constructs as theoretical concepts and modelling constructs as producers of data. To circumvent these problems, the *evaludation* process proposed by (Augusiak et al., 2014) is applied. This framework was briefly touched upon for model verification and experimentation and will be fully applied in this section.

When applying an exploratory modelling approach, such as EMA, the problem arises that strict model validation is impossible, as exploratory simulation models operate under great uncertainty (Kwakkel & Pruyt, 2013). By definition, there is virtually no tangible data available to validate model outcomes, since the model seeks to explore what would happen in hypothetical future scenarios (Bankes, 1993). This does not mean that validation steps will be skipped altogether. Augusiak et al. (2014) propose a framework to deal with uncertainty around the validity of model parameters by emphasising the validity of concepts the model was built upon. Bankes (1993) prescribes the validation process as judging model quality based on the completeness of the model – in other words, ensuring that concepts forming the foundation of the model are built on solid grounds. In that sense, traditional model validation as described by Nikolic et al. (2013) is impossible and not attempted. Instead,

evaludation serves as a thorough approach to validate conceptual elements and their translation into simulation modelling constructs. The interpretation of model concept validation comes with a major caveat: behavioural tendencies can be discussed and compared, but direct interpretability of model outcomes are meaningless.

The *evaludation* approach proposes standardised evaluation of data quality and the knowledge gaps in order to reduce uncertainty surrounding model outcomes. This is done through six steps that will be briefly discussed.

### 8.2.1 Step 1: data evaluation

The first step of evaludation is to evaluate all tangible data objects found to serve as input for the model. This was covered under model parameterisation, found in Appendix F. All model parameters were listed, assigned a base value, either based on available information or guesstimated. Afterwards, the quality of these parameters was assessed. This step helps determine the role played by uncertainty for these parameters and the outcomes were used to generate the set of scenario parameters and associated value ranges. Overall, the quality of data was relatively low for the agent-based model. This is inherently tied to the nature of agent-based modelling, especially so for exploratory purposes of an ecosystem. The concepts that were most problematic to find data requirements for were directly deduced from well-cited academic frameworks for modelling critical infrastructure ecosystems. The studies these concepts were derived from used similar approaches to formulate a proof of concept for certain conceptual relationships rather than unambiguous prediction of system states.

### 8.2.2 Step 2: conceptual model evaluation

The second evaludation step is to evaluate the concepts that form the basis of model interaction in terms of consistency. However, as Augusiak et al. (2014) remark, there is no clear-cut method for establishing how consistent model concepts are. As stated before, the overall quality of the model is based on how complete it is in terms of incorporated concepts. To achieve this, thorough literature study was conducted, as detailed in chapters 2, 3 and 4.

*Table 8-1: Behaviour and performance for each defensive strategy*

| *Strategy* | *Behaviour and performance* |
|---|---|
| *(1) Fully anomaly-based intrusion detection & prevention* | This strategy was found to be the most problematic and least robust in terms of overall performance. Conducting both intrusion detection and prevention as anomaly-based control mechanisms leads to substantially larger losses than incurred by other strategies. While effective at preventing and thwarting attacks, the undesired consequence of these mechanisms is that crucial user traffic is prevented and the overall degree of situational awareness is further clouded by false positive detections. The ecosystem is almost in a constant state of alleviation or retention with this strategy, since defenders' perception is almost constantly clouded by false alarms for non-existent attacks. This also reflects one of the main generalisations made in implementing the model, since it is virtually impossible to formalise true representations of the complex decision-making process in case a false alarm occurs. The benefits from using this strategy go hand in hand with impediments encountered. |
| *(2) Fully signature-based intrusion detection & prevention* | This strategy was found to be the overall most effective strategy for most performance indicators. Behaviour observed across the total set of experiments is also the most robust of any defensive strategy. However, there are several possibly problematic tendencies that do not directly present themselves from individual plots. This strategy is very ineffective at thwarting attacks, indicated by a relatively high number of active attacks. Despite the higher frequency of attacks, this strategy is characterised by the most efficient decision-making tendencies. While this strategy performed well across the scenario space applied for experimentation, it relies on one major assumption to perform well: that all attacks are surmountable. This strategy thrives on situations where consequences from attacks are always recoverable and clusters of nodes can operate in isolation. |
| *(3) Hybrid between anomaly-based intrusion prevention & signature-based intrusion detection* | The third strategy sought to combine the best of strategies 1 and 2. In general, performance between strategy 2 and 3 are very comparable, albeit that strategy 2 tends to perform slightly better and slightly more robust in most areas. In many cases, these differences are marginal. There is, however, one major exception to this comparison, as this strategy performs significantly better at thwarting attacks and dealing with detected attacks in a timely manner. Attacks are prevented successfully significantly more often, while detection of unprevented attacks trends almost identically. In comparison with strategy 2, the deviation in impact assessment is less problematic since attacks are prevented more effectively to begin with. While strategy 1 relies on a substantial frequency of attacks to warrant its benefits and strategy 2 relies on surmountable and uncommon attacks, strategy 3 encounters no such limitation. |
| *(4) Hybrid between signature-based intrusion prevention & anomaly-based intrusion detection* | The last strategy was another attempt to combine the best of other defensive strategies. This strategy was found to suffer from similar impediments as the first strategy, as situational awareness was clouded by frequent false alarms. This resulted in a significant number of alleviation decisions made across all experiments with this strategy active. While incorrect decisions based on overestimations were a rare occurrence, this is negligible in comparison with the high fraction of underestimations leading to incorrect decisions. |

May (2004) describes the importance of keeping model data structures simple and implicitly tied to validated concepts that form the underlying model structure. Modellers tend to overstate the complexity of certain concepts incorporated in simulation models, further increasing the sensitivity of a model towards specific values that might not always be realistic or available. To mitigate these issues, concepts identified in chapters 2, 3 and 4 were boiled down to a generalisable set of relationships correspond with an ecosystem-level perspective. To achieve this, the central degree of operability discussed in section 4.2 was applied, as it was found to work well as a facilitating factor for actions and interactions for either critical infrastructure dynamics, cybersecurity elements as well as decision-making models. Operability as a concept worked particularly well in adapting cyber-risk into an applicable concept for critical infrastructures. For example, Setola and Theocharidou (2016) prescribe a formalisation for dependency weightings based around the central degree of operability, which could then be related to the deployment of specific types of control mechanisms based on perceived operability (Sridhar et al., 2012). Another example would be the inclusion of both physical and economic loss factors, which are crucial for determining the types of attacks that take place within the ecosystem (Miller & Rowe, 2012). For an ecosystem, values assigned to these concepts mean very little – there is simply no unambiguous way to determine a set of quantitative inputs for losses associated with conceptual infrastructure nodes. Instead, in an attempt to reduce the uncertainty surrounding these parameters, the focus was shifted towards purely decision-making elements. Loss factors were merely included to facilitate target selection for attackers and the values used only indicate the relative presence of either type of consequences for nodes. Finding specific values for these parameters can therefore be considered unnecessary, as the conceptual basis for which these parameters exist was already defined.

By consistently applying this methodology for model conceptualisation and formalisation, model parameters were kept conceptually robust and relatable to instances of critical infrastructure ecosystems. Through this, the quality of the model was safeguarded by forming a complete overview of concepts required to model the system appropriately.

### 8.2.3 Step 3: implementation verification

The third step of evaludation is to verify whether the simulation model corresponds with the formalised conceptual model. This step for evaludation is fully in line with the verification steps prescribed by Nikolic et al. (2013) that were applied in section 7.5. Full results for the verification phase are denoted in Appendix E. To recap the findings discussed in section 7.5, the verification phase helped establish several implementation errors relating to the software package used to create the model. After resolving these, output was verified to assess whether observed values and distributions match expectations. The model showed a high degree of stability across time steps, showing that a sufficient number of repetitions show near-symmetrical output parameter distributions.

### 8.2.4 Step 4: model output verification

The fourth step of evaludation a model is to verify whether model output matches observations and prescribed expectations. In general, this step is similar to the overall model validation phase for the agent-based modelling cycle by Nikolic et al. (2013). However, the context of evaludation focuses on the predictability of individual model components as opposed to general simulation results. Instead, the response of model entities to changes in context parameters is assessed to establish whether patterns make sense in comparison to real-world observations. Specifying model output verification to validation of individual sets of behaviour makes validating individual elements possible. This was gradually touched upon in section 7.3 and will be further specified in this subsection. The main verifiable elements of model output relate to the process of decision-making for defenders under

different circumstances. As was identified in section 7.3, the different strategies had significant impact on how the threat environment was assessed by defenders.

Full anomaly-based intrusion prevention and detection led to severe obfuscation of whether attacks were taking place, as many false alarms were thrown. One of the core problems encountered by cybersecurity for CIs is creating an unambiguous overview of the threat environment without raising insurmountable amounts of false alarms (Patel et al., 2013). The observed emergent pattern in this instance is the almost omnipresent decision to act on false alarms throughout all experiments, leading to a substantial increase in sustained losses. This behaviour deviates significantly from the other strategies, which managed to generate more manageable impact assessments. Similar behaviour is identified in the real world, where slight deviations in impact assessment can quickly lead to wrongful defensive decisions (Department of Homeland Security, 2015). In some cases, the absence of correct threat monitoring was observed to lead to unnecessary outages, which resulted in dealing preventable damage to infrastructure assets (Clark et al., 2017). Unnecessary defensive decisions were also found to directly affect the operability of infrastructure nodes, corresponding with elements discussed by Asnar and Giorgini (2006), which were used as the foundation for responsive mechanisms.

Full signature- or specification-based intrusion prevention and detection led to robust decision-making, as the rarity of attack events outweighed the possible negative consequences predicted for such control mechanisms (Patel et al., 2013). Since there were not many attacks to account for, suppression of negative consequences from false positives was minimised. The behaviour shown by these entities is fully in line with expectations set in the context of the formalised conceptual model. However, there is one key element missing in the model that might shape the real-world threat environment: if attacks are repeatedly successful, the system would likely be subject to a higher frequency of subsequent attacks, as new attack vectors might be enabled (Department of Homeland Security, 2015). Under static circumstances, the observed behaviour is in line with expectations. Incorporating a dynamic, evolving threat landscape would be impossible within the constraints of this study, and possibly also undesired (May, 2004). For exploratory purposes, the behaviour develops as expected, but the limitations and assumptions under which patterns were observed should be noted and memorised.

The first hybrid defensive strategy, strategy 3, incorporating both signature- and anomaly-based control mechanism, showed a more robust set of behaviour patterns in dealing with higher and lower attack frequencies. Significant emphasis was placed on the importance of preventing security incidents for critical infrastructures in chapter 2. Any event where attacks are successful is undesired and should be mitigated, so long as effective decision-making is not significantly impeded. Other defensive strategies resulted in different but expected behavioural patterns in key dimensions: strategy 1 led to an obfuscation of situational awareness and strategy 2 led to subpar understanding of when attacks were taking place. Combining signature-based detection and anomaly-based prevention from both strategies suppresses both negative and positive elements from those strategies, but provides robust and less assumptive behaviour. Strategy 4, which is the inverse hybrid version of strategy 3, still suffers from problems described for anomaly-based intrusion detection.

### 8.2.5 Step 5: model analysis

The fifth step of evaludation is to analyse the sensitivity of the model to different values for certain parameters. As part of EMA, this was conducted during model exploration, the results of which are denoted in section 7.2.2. These experiments were conducted with evaludation in mind, ensuring that factors used to iterate experiments were useful for eventual validation.

### 8.2.6 Step 6: model output corroboration

The sixth and last step of evaludation is to compare model predictions with datasets and patterns not used during model implementation. This step involves partial selection of patterns observed in the model and to identify these patterns with real-world cases. Because validation using applicable data is typically nigh impossible, the number of patterns analysed is kept to a minimum. Only core patterns are compared to new sources of information, in this case relating to defensive decisions made for different incidents. Given the scarcity of information for critical infrastructure incidents, the same set of attacks recurs throughout academic literature. Several references for data will therefore have to be re-used from chapters 2, 3 and 4. The main examples with ample evidence are Stuxnet and the 2015 Ukrainian blackout (Farwell & Rohozinski, 2011; Karnouskos, 2011; Lee et al., 2016; Liang et al., 2017). To corroborate behavioural tendencies, these examples and several smaller examples will be used. Model experimentation discussed in section 7.3 can be boiled down to three main behavioural tendencies:

1 When there is no sufficient awareness of attacks, the impact of those attacks is amplified as the deployment of responsive mechanisms/defensive decisions is delayed. Real-world examples of attacks that go undetected for prolonged periods show the severe extent of damage that can be inflicted, as such this behaviour might not be desirable, even if the false alarm rate is negligible.
2 When there is an unmanageable rate of false alarms, responsive mechanisms made can end up inflicting more damage than simply doing nothing. Defensive strategies might perform well at thwarting attacks in a timely manner, but might ultimately end up damaging the ecosystem more than attacks would.
3 Preventing attacks by design delivers robust performance across simulations whereas preventing false alarms bears the most immediate benefits, since attack events are relatively uncommon. These strategies avoid overly relying on assumptions for cybersecurity scenarios.

The first tendency is the most prominent element discussed in literature on cyberincidents for critical infrastructures: insufficient attack detection or response protocols leading to substantial damage as attacks can inflict damage for prolonged durations. The biggest example of such an attack is Stuxnet, which managed to infiltrate systems for a long time without making any significant deviations (Karnouskos, 2011). After a prolonged time, the worm would take over control of physical equipment and destroy facilities. Failures to accurately assess impact could be problematic for real-world scenarios where attacks are insurmountable and deal lasting damage. These types of attacks are essentially impossible to include in a simulation model, as they are incredibly rare occurrences.

The second tendency can be corroborated by several smaller examples. Clark et al. (2017) discuss such an example, where a water and wastewater facility in Boca Raton, USA reacted to cybersecurity incidents inappropriately, causing control systems to damage itself. Instead of implicitly removing perceived threats, the control system caused further malfunctions trying to remove a non-existent threat. Due to insufficient monitoring of the threat landscape, harmful defensive decisions were made, substantially damaging system operations. Another example denoted by Department of Homeland Security (2015) involves Newark Airport operations preventively being taken offline after a suspected cyberattacks was perceived. In reality, a truck driver using a commercial-grade GPS jammer passed by the airport.

The third tendency reflects on the robustness of performance across assumptions made for the simulation model. The attacks on the Ukrainian power grid in 2015 incorporated multi-faceted attack vectors to infect systems with malware that could be used together to inflict physical damage (Karnouskos, 2011). Individual system elements might have detected attacks but failed to account for the danger posed by coherent attack vectors. This underlines the notion discussed in section 7.3.2

that relying on the assumption of relatively surmountable attacks is unlikely to yield desired and robust results for real-world analysis. These situations could have been prevented upfront by including security requirements by design rather than loosely-coupled security requirements (Fairley, 2016; Neuman, 2009).

### 8.2.7 Model validity assessment

After iterating through the six evaludation steps and critically reflecting on observed emergent patterns, it becomes clear that attempting to validate exploratory models is a complex endeavour. However, by iterating through a standardised evaluation scheme, uncertainty surrounding model parameters and resulting patterns observed within the set of simulations could be suppressed. By drawing a parallel between observed patterns and historical events analysed by academics, a certain degree of corroboration could be formulated. This is sufficient for stating that the model can be used to predict possible system behaviour, specifically assessing what happens with critical infrastructure elements in an ecosystem built around cybersecurity concepts.

## 8.3 Intermediate findings

This chapter continued from chapter 7, further analysing data generated from experimentation and ensuring the validity of findings. Section 8.1 detailed robustness analysis across designs for defensive strategies, denoting the emergent patterns observed. It was found that there were several delicate differences in decision-making processes that emerged from different circumstances. It was shown that defensive strategies significantly influence the robustness of system performance across a large set of scenarios. No strategy was dominant, but simulating all strategies helped generate insight into what happens in the ecosystem of critical infrastructures. Traditional model validation proved to be impossible, as values applied in the model are sometimes arbitrary and in many cases assumptious. The evaludation approach for exploratory simulation models helped establish the conceptual validity of modelling constructs, in an attempt to validate the foundation on which data analysis was conducted. While results deducible from model simulations can be formulated, significant consideration will need to be applied to how assumptious any findings might be. These findings serve to help understand behaviour that might occur under deep uncertainty and cannot be used to argue for a specific defensive strategy based on statistical performance observations.

# 9 Conclusion and discussion

After conducting all required phases of research, the main findings can be wrapped up into coherent answers to all research sub-questions and eventually the main research question. First, the limitations to the overall study are addressed. Secondly, the main findings are described, formulating answers to all research questions. Thirdly, the implications of the findings are discussed. Fourth and last, options for future research are discussed.

## 9.1 Limitations

Before the conclusions and main findings can be laid out, the limitations to these findings should be described. There are multiple facets that played a role throughout this study that could limit the representativeness of the model and subsequent findings. It is important to discuss possible limitations to ensure possible uncertainties are accounted for, even if all steps for 'proper' model development were conducted and consistently documented.

### 9.1.1 Limitations of an ecosystem model

The most influential set of limitations discussed arises from the ecosystem-level approach to critical infrastructures. The knowledge gap established through literature review, discussed in Appendix A, specifies the need for coherent security policies that incorporate cyber-risk as a shared property of the entire ecosystem. For this reason, the ecosystem was designed around properties for critical infrastructures that describe how reduced operability from single infrastructure nodes transcends beyond that node. Examples of these properties are implicit dependencies between two nodes or the degree of situational awareness that constrains decision-making. While there is a shared desire to explore ecosystem-level behaviour and assess the effects of different types of defensive strategies, this approach is not without its limitations.

*Generalisability of findings*

The first major problem with ecosystem-level models is that they should incorporate elements that are generalisable to most, if not all, applicable critical infrastructure systems. This requires selection and abstraction of elements that should and should not be incorporated, leaving out other elements that define individual infrastructures. Elements that are incorporated should be modelled in such a way that they are applicable to other critical infrastructures. This inherently reduces the complexity of incorporated concepts, which aids the modelling process, but might hurt the representativeness of the eventual simulation model. There is a delicate balancing act between creating models that are too sensitive to yield generalisable insights for other infrastructure sectors and models that are too generalisable that no real value is added. Fortunately, the results from experimentation showed variability and emergent behaviour for several concepts. On the other hand, the actual meaning and interpretation of these results in relatively uninspiring, as they cannot be translated directly into desired policy requirements. Whether or not the model in its current state can be utilised for research on specific defensive strategy implementations while incorporating more specific, infrastructure-related elements is uncertain. The original aim for this model was to be based around a specific type of infrastructure to ensure tangible results. This was turned around in favour of an ecosystem-level approach at an early stage of this study, as the assumptions required to model such a system given the specified time and resources would water model concepts down to meaningless interpretations. This made sure that results would be too abstract for direct interpretation, but it was also a crucial and necessary first step towards creating models that can serve that purpose.

*Lack of tangible data as model input*

The second major problem for ecosystem modelling was encountered during model formalisation and implementation: the lack of tangible, representative data objects to serve as for model input.

When analysing a single infrastructure, specific and realistic data can be found, paving the way for specific, near-deterministic simulation models. Creating such a simulation model provides more straightforward interpretation of model outcomes, but makes the model more sensitive towards domain-specific properties. This raises requirements for the quality and coherence of data used, but also enables more advanced research into designs. The decision to create an ecosystem-level simulation model constrains the direction of research to exploratory purposes. Since the desired direction of research is in line with the possibilities and impediments of the chosen approach, this is not problematic, so long as the choice was a conscious decision. In fact, creating such a model was also the only option given the current state of the art, time and resources available for this project. The framework created as part of this study can be used for further research that incorporates tangible, well-supported behavioural mechanisms and statistical data to ensure specificity of outcomes is not tarnished by abstraction of model constructs.

### 9.1.2 Limitations resulting from the CAS perspective and agent-based modelling

Complex adaptive systems thinking perceives systems as complex collections of entities operating based on basic states, rules and actions. This perception ensures a coherent and interconnected ecosystem that enables operationalisation of interaction. However, this also results in several limitations.

*Abstraction of interaction*

The first issue is that not all concepts related to cybersecurity for critical infrastructures can effectively be boiled down to a set of basic interactions. Examples of these are complex social or economic drivers behind cyberattacks towards specific types of infrastructures, geographical locations of infrastructure nodes affecting populous areas, or the complex decision-making processes for defensive decisions. An even more prominent example relates to the behavioural models used for attacker and defender behaviour. There is no single best option to model how resources are allocated, how attackers select targets and how defenders respond to intrusions. The simulation model created as part of this study largely skips game theoretic dilemmas, as attack operation is handled probabilistically and defensive strategies are part of static system configurations. Since these game theoretic elements were not all included, possible differences between sets of behavioural rules and actions or the effects of strategic behaviour were not fully explored. However, implementing some of those concepts and relationships without incorporating their root causes would make the model even more assumptious by design, as the presence of most root causes cannot be unequivocally determined. To this end, keeping the conceptual framework narrow can help pinpoint emergent patterns in behaviour at the cost of being able to determine direct consequences for certain system configuration. Instead, research was forced to be tailored more around exploring different scenarios and analysing the robustness of system configurations rather than determining their direct performance. This proved to be sufficient in generating a model and framework to serve as a proof of concept for ecosystem-level modelling.

*Ambiguity of behavioural patterns*

The second issue with perceiving cybersecurity for critical infrastructures as a complex adaptive system is that the CAS framework revolves around detecting emergent patterns and self-organisation. These patterns are observed as emergent from the collection of entities included in the model. Cyberattacks on critical infrastructures are found to be ever-evolving in a way that cannot unambiguously be captured within CAS thinking. Patterns observed in the real-world are sensitive towards several factors that cannot be modelled in a straightforward manner. The desired observed patterns in an ecosystem of critical infrastructures are ideally rather subtle in nature, as large-scale disruption and self-organisation is the result of an evolving threat landscape and lacking defensive

strategies. Instead, a level of ambiguity was applied to explore the response of the system to certain shifts in environmental drivers. This is part of the aforementioned balancing act between meaningful and generalisable models, a key issue being that the total set of modelling choices and assumptions should be coherent.

### 9.1.3 Limitations of the model and exploratory modelling

The most prominent limitations to any agent-based model are formed by the usage of assumptions for model concept formulation and ensuring elements included in the model are coherent. The main modelling assumptions applied for concept formalisation are denoted in appendix D. For this study, many such issues were circumvented by ensuring the concepts incorporating in the model were based on a solid foundation of academic literature.

*Validatability of the simulation model*

Conceptualisation was conducted with difficulties in validating the eventual simulation model in mind. Examples of such model concepts are the dependency weighting model by Setola and Theocharidou (2016), which in itself is related to an extent of operability that can be expressed as a numerical value, corroborated by Puig (2018), or the connection between intrusion prevention and detection events and a degree of situational awareness for this level of operability. By ensuring the concepts that served as input for the model are connected, no major assumptions had to be made with regards to core interaction, fulfilling part of the validation process. Instead, the engineered framework covered aspects that were previously not all connected. However, there are still numerous assumptions that impact model behaviour, such as the assumptions made to be able to implement crucial user traffic as a modelling concept. This inclusion was required to match actions for control mechanisms and the level of operability, but besides the existence and actions of this traffic, there was no clear guideline for the exact frequency and extent of this interaction. In order to keep the model manageable, several assumptions were made to make implementation possible. These assumptions relate to the frequency at which intrusion detection takes place and the relative frequency and criticality of user traffic for infrastructure node operation. These assumptions are not inherently problematic, as they are in line with value ranges specified for the ecosystem model and multiple value ranges were iterated over as scenarios for experimentation. Evaludation proved valuable in establishing whether concepts were implemented in a thorough and representative manner.

*Abstraction of model concepts*

As addressed by the limitations for ecosystem models and the CAS perspective, selection and abstraction of concepts was required, indicating that several conceptual factors were originally found but pruned from the conceptual model. The original research proposal included a distinction between multiple types of critical infrastructures based on hierarchical structure. The idea behind this was to analyse different types of systems interlinked by cross-sectorial dependencies, assessing how resilient and robust each type of infrastructure would be given defensive different strategies. This was ultimately decided to be unnecessarily complex, as it would require both a significant understanding of implicit specifics as well as increased reliance on tangible data from each infrastructural sector. Instead, the model was kept manageable by including one conceptual network of infrastructure nodes. This performs better on an ecosystem-level, and while the objectives for the model are different, this approach was more likely to successfully yield useful results.

*Limited meaning for policy design without specification*

Since the conscious decision was made to not analyse individual or multiple specific critical infrastructures, the implications for data input and output were also clear: there would be little to no realistic data available for meaningful implementation in the model, and as a result there would be

little meaningful interpretation of values for performance indicators resulting from simulations. A solution for this limitation was found by following the EMA framework, which helped reduce the uncertainty for model outcomes by thoroughly varying across values for uncertain scenario parameters. Instead of focusing on direct interpretations for values of performance indicators, analysis focused on observing the robustness of several design alternatives across the scenario space. To this end, data analysis was also limited to exploratory analysis of distributions for model output parameters across the set of experiments. This proved that, even with several parameters using meaningless values, valuable insights in system behaviour could be found, generating key insights required to answer research questions. The generalisability of results was not tarnished, since a methodical, thorough approach was taken to experiment with unvalidatable models. However, the actual meaning of results was doubtful, as stated before, as it followed circular reasoning. Information on control mechanisms was used as input for the model and was corroborated by model output. In essence, no new insights were gathered through this process alone. Despite reducing uncertainty surrounding the validity of model behaviour and model output through evaludation, the findings discussed throughout this study bear few direct implications for policy design. Instead, the model was used to explore what happens if a hypothetical coherent defensive strategy was implemented across the full ecosystem. More specifically, the study functions as a proof of concept that this modelling approach can yield desired behavioural patterns and can be extended with specific details for specific infrastructural sectors for policy design or requirements engineering.

## 9.2 Main findings

After addressing the main limitations to this study, the main findings can be formulated. First, these will be synthesised in a set of answers and discussions for research questions. Next, the main contributions taken from this research are discussed.

### 9.2.1 Answers to research questions

With all research steps now conducted, findings can be formalised into answers for all research questions. The research questions formulated in chapter 1 are recalled in Table 9-1, along with initial research objectives. Each question will be addressed individually and in chronological order.

*Sub-question 1*

The first sub-question revolves around identifying architectural elements that define the resilience of critical infrastructures against cyberattacks. The insights required to answer this sub-question were gathered and discussed in chapter 2, tailored around mapping architectural complexities and their impact on critical infrastructure operation. It was found that there are four major components that define this impact:

1. Vulnerabilities that arise from the use of networked heterogeneous systems, as the multitude of different sensors and subsystems complicate security practices.
2. Dependencies between infrastructure nodes causing cascading failures. These determine the perturbation effectuated on dependent nodes, caused by a disruption in the origin node.
3. Extending dependencies, the networked structure of critical infrastructures determine how inoperability in a node causes disruption in further nodes. Depending on the networked structure of a CI system, this might result in isolated incidents or widespread inoperability.
4. Severe consequences from critical infrastructure inoperability that strengthen the need for effective security approaches, as any slight disruption could set a destructive chain of events in motion.

*Table 9-1: Research questions and objectives*

| *Question/objective* | *Formulation* |
|---|---|
| *Sub-question 1* | *How does architectural complexity of critical infrastructure nodes within the cybersecurity ecosystem affect infrastructure operation?* |
| *Sub-question 2* | *How do control mechanisms and cyber-threats secure or impede operation of critical infrastructures?* |
| *Sub-question 3* | *Which properties for attacker and defender behaviour aptly describe decision-making behaviour in the cybersecurity ecosystem of critical infrastructures?* |
| *Sub-question 4* | *Which emergent behavioural patterns can be observed in interactions within the cybersecurity ecosystem for critical infrastructures?* |
| *Sub-question 5* | *What can be learned about the effectiveness of defensive strategies with regards to robustness and resilience in the cybersecurity ecosystem for critical infrastructures?* |
| *Main research question* | *How do cyber-architectural elements and defensive strategies influence exposure to cyber-threats within the cybersecurity ecosystem of critical infrastructures, and how can infrastructure operators effectively mitigate consequences from cyber-incidents?* |
| *Objective 1* | *Specify and conceptualise an ecosystem model for CI systems by establishing elements that relate to each core concept.* |
| *Objective 2* | *Formalise and specify this ecosystem into a fully-fledged agent-based model capable of simulating different configurations for integrated defensive strategies.* |
| *Objective 3* | *Derive the simulation results into emergent patterns and best practices for effective cyber defensive strategies for critical infrastructures.* |

*Sub-question 2*

The second sub-question required specification of elements related to cyber-incidents. The required insights were gathered and detailed in chapter 3, which aimed to establish how cyberattacks take place within the ecosystem, as well as control mechanisms in place to thwart cyberattacks. It was found that cyberattackers and cyberdefenders are both characterised by three elements. For attackers, these elements are:

1. Attacker types representing the nature of an attacker. Cyberattackers targeting critical infrastructures are generally only considered as advanced persistent threats, indicating that the types of attacks launched towards critical infrastructures tend to be coordinated, impactful events.
2. Motivations for attackers determining the type of consequences preferred by different types of attackers. This impacts the type of attacks they would employ. Cybercriminals are less likely to use attacks as a means of inflicting physical harm than cyberterrorists, as they mainly prefer economic incentives.
3. Attacker capabilities that describe the capabilities for each type of attacker, indicating the means available and typically used by types of attackers. Foreign adversaries are more resourceful than other attackers, as they are capable of developing highly specific, targeted attacks.

For defenders, the following elements were identified:

1. Control mechanisms being used by defenders to *prevent*, *detect* and *respond to* attempted intrusions by cyberattackers. These impact the success rate of cyberattacks, as well as the timeliness by which attacks are detected and responded to.
2. Defensive strategies comprising of a combination of control mechanisms, essentially serving as the set of behavioural rules for defender agents.
3. Infrastructure operators acting on a degree of threat awareness that impacts the usage of several control mechanisms. Defensive decisions are made based on the perception of the threat landscape. Critical infrastructures deviate from traditional cybersecurity paradigms, where security investments are based on perceived costs and benefits over time. The severe consequences imply that threat awareness only determines defensive decisions for CI systems.

*Sub-question 3*

The third sub-question sought to establish a set of properties that impact decision-making behaviour observed by cyberattackers and critical infrastructure operators. Three concepts were identified and discussed:

1. Situational awareness for attackers determining their targeting selection behaviour. Attackers operate on an established degree of available knowledge by which they assess which infrastructure node would be their optimal target.
2. Infrastructure node operability as a scale central to interaction within the model. Node operability is affected by cyberattacks, dependencies and handling of user traffic and also enables threat awareness to be implemented as a perceived degree of this scale.
3. Situational awareness for infrastructure operators, represented by the effective perceived level of operability of an associated node. Infrastructure operators use this degree of situational awareness to make decisions and assess whether their situational awareness should result in a responsive mechanism being used.

*Sub-question 4*

After thorough analysis and discussion of behaviour across the complete set of experiments, three main patterns were identified. Together, these patterns describe the main deviations occurring throughout model runs and show the dynamicity of system configurations responding to environmental drivers. These patterns are formulated as follows:

1. When there is insufficient awareness of threats, the impact of those attacks is amplified as the deployment of responsive mechanisms/defensive decisions is delayed. Real-world examples of attacks that go undetected for prolonged periods, such as Stuxnet, show the severity of damage that can be inflicted (Farwell & Rohozinski, 2011; Karnouskos, 2011). As such this behaviour might not be desirable, even if the false alarm rate is negligible by metrics used in the simulation model.
2. When there is an unmanageable rate of false alarms, responsive mechanisms made can end up inflicting more damage than simply doing nothing. Defensive strategies might perform well at thwarting attacks in a timely manner, but might ultimately end up damaging the ecosystem more than attacks would.
3. Preventing attacks by design delivers robust performance across simulations whereas preventing false alarms bears the most immediate consequences, since attack events are relatively uncommon. These strategies avoid overly relying on assumptions for cybersecurity scenarios.

*Sub-question 5*

After analysis of observed behavioural patterns across different defensive strategies and scenario parameters, the following statements can be derived:

1. Defensive strategies that incorporate anomaly-based intrusion prevention manage to thwart attacks and their direct impact much more effectively than defensive strategies that make use of their signature-based counterparts. On the other hand, strategies with signature-based intrusion prevention ensure defensive decisions are made more accurately, since problems caused by falsely blocked user traffic occur less often.
2. Defensive strategies that make use of anomaly-based intrusion detection further suppress the impact of attacks, as they are dealt with marginally more quickly. The larger impact made by intrusion detection mechanisms relates to the effects on defensive decision-making correctness. Strategies using anomaly-based detection mechanisms result in serious obfuscation of situational awareness, since false positives occur more frequently than false negatives due to the inherent rarity of cyberattacks. Within the model, the effects of unnecessary defensive action outweigh the effects of lacklustre attack mitigation.
3. The overall best performing defensive strategies (strategies 2 and 3) both incorporated signature-based intrusion detection and therefore avoided the complete obfuscation of situational awareness. However, these aforementioned patterns describe the balance that follows the implementation of abstract and conceptual control mechanisms without incorporating specific elements of decision-making processes or intricacies that affect the use of control mechanisms. Instead, it is important to take note of emergent behavioural tendencies and relate these to other modelling constructs. Because the model is inherently unvalidatable, the meaning of statistical values is useless for further analysis of ecosystem-level behaviour.

*Main research question*

The main research question is two-tailed: the first half of the question seeks to establish understanding of how conceptual elements influence system behaviour, whereas the second half revolves around understanding how future system behaviour could be predicted based on those conceptual elements.

It was found that by studying the cyber-architectural complexity of this ecosystem, the mechanisms that enable and thwart attacks and implications for interaction, a solid basis of understanding was created. In order to generate new insights and contribute to the academic state of the art, a new direction was taken, as these concepts were operationalised on the scale of a conceptual ecosystem. To achieve this, a framework for operationalisation was required that links all previously identified concepts. This framework would then pave the way for translation into modelling constructs for simulation-based analysis. By relating all identified concepts to a central element of the ecosystem, these concepts could be connected in a coherent and thorough framework. This central element is infrastructure node operability. Operability can be used to describe the effects of defensive actions, offensive actions, dependencies and interdependencies as well as enabling a simplified interpretation of decision-making processes. In doing so, several concepts were simplified into an abstract representation on an ecosystem level. This implies that quantitative outcomes are less insightful, but made sure that simulation modelling was possible on this level.

To answer the second part of the question, emergent patterns and key insights resulting from the simulation model were used. By exploring possible deviations and instances of coherent defensive strategies, the relative robustness of conceptual defensive strategies could be analysed. The exploratory nature of this study implies that results do not directly provide insight into the

effectiveness of tangible design alternatives. Because defensive strategies were abstract representations of conceptual arrangements of control mechanisms, the implications resulting from model runs themselves are rather minor. Emergent patterns found as part of sub-question 5 shed some light on expected behavioural patterns possible in the model, but fail to unequivocally produce new insights that were not implied by the concepts model elements were based on. This is largely attributable to the level of abstraction used, as the results cannot be used to support the results without engaging in circular reasoning. Instead, the implications of this research relate to the proof of concept for ecosystem-level modelling as a means of simulating the way concepts interact. Since the model itself is unvalidatable, conceptual validation was applied through evaludation to ensure that behavioural patterns made sense. It is important to keep in mind that for any modelling study, the outcome is an artefact of the applied set of assumptions. This paves the way for future research to extend or apply this framework for critical infrastructure ecosystems in a more advanced setting to produce specific, tangible results. While this is the desired goal to generate new knowledge in addition to the academic state of the art, this was not possible without formalising and operationalising a coherent ecosystem-level framework.

### 9.2.2 Main research contributions

Having answered the research questions and fulfilled the research objectives, the main insights gathered throughout this study have been noted. This subsection will discuss the main academic and societal contributions provided by this study. Recalling the knowledge gap, the aim of this research project was to enhance knowledge on the ecosystem-wide effects of certain defensive decisions. Each contribution will be described in the light of the academic state of the art.

*Ecosystem-level aggregation of critical infrastructures*

The first major contribution provided put forth by this study is an aggregation of concepts incorporated in an ecosystem model for critical infrastructures. This ecosystem-level conceptual model is the result of achieving the first research objective and contains all generalisable concepts that together represent an ecosystem of critical infrastructures. The knowledge gap formulated in chapter 1 specified the need for analysis of ecosystem-level interaction and the effects of cyberattacks and defensive strategies on this ecosystem. However, no coherent framework or other integration of cybersecurity elements of critical infrastructures was put forth. This ecosystem-level aggregation identifies and conceptualises such elements and how they are interrelated. The model is depicted in Figure 9-1 and shows the specification of concepts discussed throughout this study. The core concept of infrastructure node operability is depicted centrally within this aggregation, depicting how all other elements relate back to infrastructure nodes, the central entity among interaction. In its most meaningful sense, this framework can be considered an artefact that encompasses the gaps of knowledge identified in the academic state of the art. Combining these factors and concepts in an operationalised model can be further synthesised with real-world cases to fulfil specific research objectives.

*Ecosystem interaction model for critical infrastructures*

The second contribution expands on the ecosystem aggregation model and specifies the main interactions that take place within this ecosystem, as well as operationalising concepts specified in the aggregation model. While the aggregation model provides a set of concepts that describe states and interactions required for eventual implementation of a simulation model, the interaction model provides a conceptual overview of interactions to be simulated. This model is shown in Figure 9-2 and was used as the foundation for agent-based modelling concept implementation. Avenues for operationalisation were discussed in chapter 6 and proved that despite the high-level and abstract nature of concepts in the aggregation model, elements could still be computed in a coherent

manner. This step and contribution was a key part of addressing the academic knowledge gap, as these activities are required before simulation models can be devised. Given the level of abstraction required for ecosystem-level analysis, the framework introduces a novel approach to gathering missing knowledge, and in itself is also an artefact of knowledge. The main contribution by this framework is offering a ready-to-implement ecosystem overview and a set of constructs that could be used for simulation modelling.

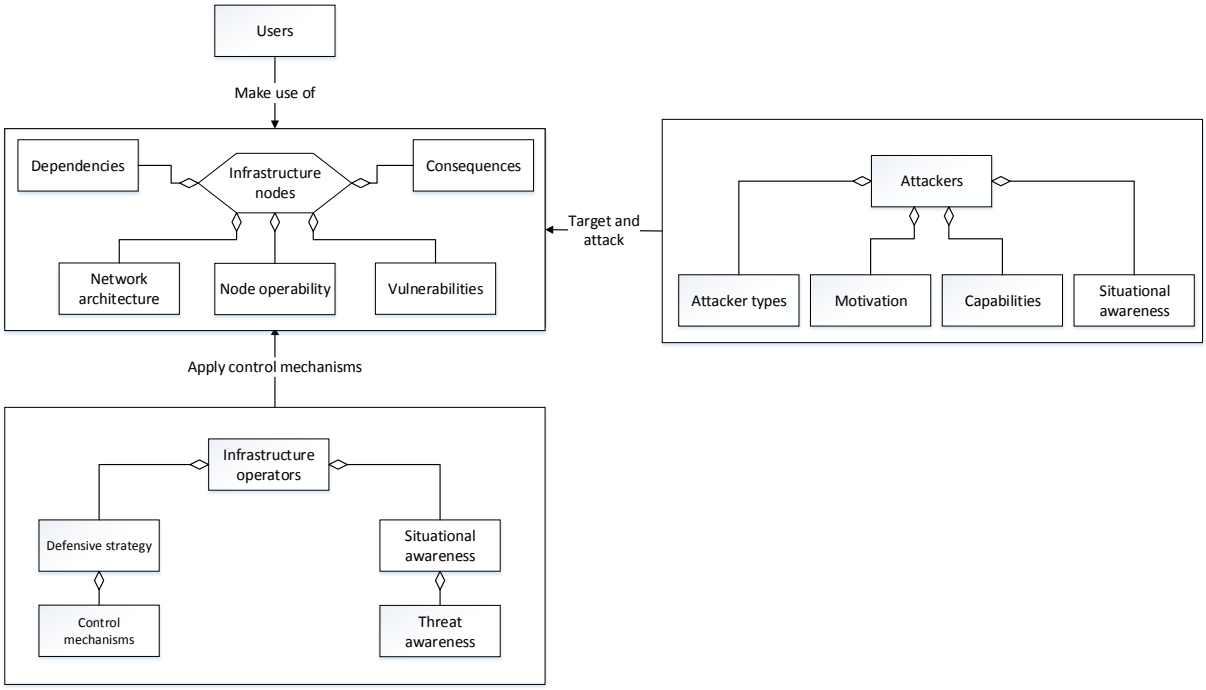


*Figure 9-1: Ecosystem aggregation model of cybersecurity for critical infrastructures*

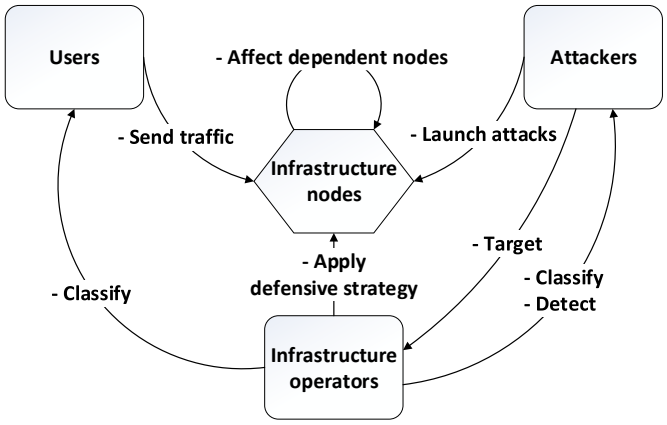


*Figure 9-2: Ecosystem interaction model of cybersecurity for critical infrastructures*

### *Explorative agent-based model for conceptual defensive strategies*

The third major contribution is the implemented agent-based model itself, as well as, to a lesser extent, the derived observations about different conceptual defensive strategy designs. The agent-based model is fully verified and validated to the extent an explorative model can be validated.

Within this model, users of the model can shift parameter values to desired configurations, with possible modules to be extended to feature more specific representations of real-world infrastructures. The model also serves as a proof of concept for ecosystem-level simulation of critical

infrastructures, showing how malleable emergent patterns can be achieved by varying between different configurations for defensive strategies.

The results from experimentation with conceptual implementations of different defensive strategies are denoted in Table 9-2. This shows for each combination of control mechanisms analysed what the observed behaviour tendencies are. Together, these observations indicate which properties of defensive strategies could result in desired performance. The full set of patterns is discussed in chapter 8 and was summarised in section 9.2.1.

*Table 9-2: Observed patterns for combinations of control mechanisms*

| *Intrusion prevention* / *Intrusion detection* | *Anomaly-based* | *Signature/specification-based* |
|---|---|---|
| *Anomaly-based* | This strategy results in the lowest susceptibility to cyberattacks, as anomaly-based control mechanisms find most attacks quickly. However, the strategy also resulted in the least robust overall performance, as situational awareness was consistently obfuscated by false alarms. Responsive measures are almost constantly conducted due to this obfuscation. | This strategy led to the overall lowest number of correct defensive decisions, as the majority of alleviation decisions were based on underestimating operability. However, together with the fully anomaly-based defensive strategy, this strategy resulted in the highest attack detection rate. Translating these findings into real-world guidelines would require further specification of decision-making processes. |
| *Signature/specification-based* | This strategy showed low susceptibility to cyberattacks, as the anomaly-based prevention mechanism manages the thwart the majority of attempted attacks. By using signature-based intrusion detection, additional overestimation of the threats posed to the ecosystem are not exaggerated, and the loss in sensitivity for detecting attacks is compensated by initially preventing the majority of attacks. | This strategy ultimately shows the most robust performance across most performance indicators, but rests on the assumption that incurring several attacks is surmountable, as signature-based control mechanisms fail to detect as many attacks as anomaly-based mechanisms. Other strategies experience most losses by making incorrect defensive decisions, which this strategy does not show. |

In the abstraction of these patterns and strategies lies the main problem with the contribution: it does not provide any tangible findings besides emergent patterns that were largely implied by the assumptions on which the model was based. Instead, the agent-based model serves as a proof of concept for operationalising the framework put forth by this paper. In this capacity, the model proves that the framework is capable of generating behaviour that corresponds with real-world observations, despite a high-level view. In order to generate tangible artefacts of knowledge to contribute to the academic state of the art, further research is required. As such, this contribution paves the way for further simulation studies in the light of the scientific gap of knowledge, much like how the other major contributions enable further research.

## 9.3 Implications

This section details how the main contributions from this study affect the scientific state of the art and what the societal implications are for these contributions.

### 9.3.1 Modelling an ecosystem of critical infrastructures

The main implication asserted by this study is evidence that an ecosystem approach to modelling cybersecurity for critical infrastructures is possible. During a comprehensive review of contemporary academic literature, the academic and societal desire for ecosystem-level analysis of critical infrastructures emerged. Following major cyber-incidents for critical infrastructures around the world, many policymakers stressed the need for new approaches to securing critical infrastructures (Fairley, 2016; Farwell & Rohozinski, 2011). Karnouskos (2011) stressed the desire for next generation control systems to incorporate coherent security policies by design. Despite this desire, there is still a gap of knowledge related to the effects of cyberincidents in critical infrastructure systems with cyber-architectural elements, rationality, dependencies and defensive strategies accounted for.

In order to contribute to the design of such policies, a framework for ecosystem-level interaction is required. This ecosystem establishes a foundation for just that, as it incorporates elements that should be included in any model for critical infrastructure systems. The model and modelling approach suggest that emergent patterns can be discovered through agent-based modelling and show that this approach is capable of exploring possible emergent behaviour. Since the theoretical implications of the direct results of this simulation modelling study are minimal, it is important to keep in mind that this modelling approach creates a foundation that could be used for future analysis of specific cybersecurity scenarios or policies. The model itself is an artefact of assumptions to demonstrate how the framework is capable of operationalising and simulating the subject matter. The framework in itself is less assumptious and specifies a coherent collection of concepts and how these factors could be enumerated.

While ecosystem-level models cannot produce results that directly translate into tangible policy implementations, they provide additional insight for the bodies of knowledge upon which future policy can be designed (Nikolic & Kasmire, 2013). Understanding how inoperability translates from one node to another is crucial to understanding the entire threat landscape (Baiardi et al., 2006). The immense consequences from critical infrastructure security incidents should be sufficient reason for creating a complete overview of the threat landscape. However, all possible future use should still respect the limitations associated with an ecosystem modelling approach.

### 9.3.2 Exploring security policies and designing future critical infrastructures

The increased frequency and impact of cyber-incidents stresses the need for coherent security policies (Brown et al., 2006; Li et al., 2012; Neuman, 2009). Historically, cybersecurity requirements for critical infrastructures were often an afterthought, with little consideration of possible cascading failures resulting from cybersecurity practices. This study proved the possibilities to analyse the effects of security policies using an agent-based model, which can be used to test several elements that might matter for coherent security policies. Hahn et al. (2013) highlight the desire to test security elements in an architectural ‘cyber-physical security testbed’. This requires exploration of specific instances of design elements, with several additional modifications to the ecosystem model taken into account.

The translation of conceptual patterns and conceptual defensive strategy designs requires a different environment than an agent-based modelling suite, especially so when analysis is conducted on the level of the cybersecurity ecosystem for critical infrastructures. The findings discussed in section 8.2

should be taken for what they are: pure exploration of possible behaviours based on a substantial basis of core elements, and not analysis of optimal strategies to base policy on. Translating observed patterns into policy opportunities requires integrating many additional social and technical facets, while also having to account for responsible costs of defensive strategies. This step exceeds the scope of this study, but can make use of the findings and contributions discussed throughout this chapter. While coherent, top-down policies might make sense when designed from a blank slate, they are based on a selection of properties that could be managed within a simulation model. Real-world policy implementations cannot make these assumptions and exclude properties from the applied scope.

However, this does not mean that simulation models of this type are useless for engaging the policy process. The more time passes by without tangible actions towards new, secure-by-design infrastructure, the more important complete exploration of all possible effects related to critical infrastructures is. If critical infrastructures and policymakers are to keep up with the ever-evolving threat posed by cyberattackers, they should explore all avenues possible by assessing the robustness of both tangible, domain-specific designs, as well as higher level behaviour of generalisable infrastructures. The process of designing new infrastructures is expensive, and the short-sightedness shown for legacy infrastructures should serve as a lesson for new designs: to account for possible security issues at all stages of development (Department of Homeland Security, 2015; Neuman, 2009). The main implication from this study is therefore the conclusion that it is possible to model and simulate the impact of cyberattacks and defensive strategies on interdependent networks of critical infrastructures. To achieve this, the framework created forms a solid foundation that is both a coherent integration of concepts as well as being extensible with additional concepts.

## 9.4 Future research

As stated in section 9.3, there are several possibilities to conduct further research to gather more insights into cybersecurity elements for critical infrastructures. Three main avenues for further research are encouraged.

The first possible avenue for future research is to use the simulation model designed as part of this study as the starting point for additional elements for ecosystem-level analysis. Other modellers can extend the model to add additional elements that are identified for relevant policy exploration. If this is done, the agent-based modelling cycle should still be adhered to, implying the necessity for further verification and validation. Section 9.1 detailed several possible extensions that were cut during conceptualisation of this simulation model, including the differences between different architectural compositions of the simulated infrastructure network. One particular consideration for extension is to incorporate a dynamic threat landscape that represents the probable increase in threat frequency if many incidents occur in a recent time period. Such changes can extend the range of uses for the model to also include time-sensitive developments. Another avenue for model extension is by including different decision-making models, or changing the baseline of interaction from probability-based to game theory. Including more game theoretic elements could shed light on how strategic behaviour possibly shapes system behaviour. Other researchers might be interested in expanding the model with concepts that they deem necessary to achieve their desired research objectives. However, this avenue for future research should stick to the exploratory nature of this model, and is likely mainly relevant for exploring new, untouched directions for defensive strategy designs.

The second avenue for further research also builds upon the foundation of this simulation model and involves specification of the system-of-interest. If this is done, the researchers should keep in mind that the foundation of the model was built around an ecosystem-level aggregation of incorporated elements. Shifting these elements towards specific implementations and catering towards real-world

data is possible, but might require substantial modification of procedures, such as attack selection or impact assessment. However, the underlying foundation for operationalisation should stay the same, as this was based on a thorough study and conceptually validated through evaludation. Applying such an approach can help generate new directions for security policies and formulate a more advanced stage of research than presented by this study.

The third possible avenue for further research builds upon the concepts included for ecosystem-level modelling of critical infrastructures. As denoted in section 8.2 and 8.3, the desire for understanding the effects of dependencies and situational awareness on infrastructure operation is an emerging discipline of science. To this end, the findings of this thesis are not inherently relevant and the approach might be the more interesting facet. Mimicking the process of incorporating concepts for elements of a cybersecurity ecosystems and translating these towards formalised, tangible computations of parameters can help formalise similar models for other topics.

# References


Abrams, M., & Weiss, J. (2008). Malicious control system cyber security attack case study–Maroochy Water Services, Australia. *McLean, VA: The MITRE Corporation*.

Alcaraz, C., & Lopez, J. (2013). Wide-Area Situational Awareness for Critical Infrastructure Protection. *Computer, 46*(4), 30-37. doi:10.1109/MC.2013.72

Amin, S., Litrico, X., Sastry, S., & Bayen, A. M. (2013a). Cyber security of water SCADA systems—Part I: Analysis and experimentation of stealthy deception attacks. *IEEE Transactions on Control Systems Technology, 21*(5), 1963-1970.

Amin, S., Litrico, X., Sastry, S., & Bayen, A. M. (2013b). Cyber security of water SCADA systems—Part II: Attack detection using enhanced hydrodynamic models. *IEEE Transactions on Control Systems Technology, 21*(5), 1679-1693.

Aradau, C. (2010). Security that matters: Critical infrastructure and objects of protection. *Security Dialogue, 41*(5), 491-514.

Ashok, A., Hahn, A., & Govindarasu, M. (2014). Cyber-physical security of wide-area monitoring, protection and control in a smart grid environment. *Journal of Advanced Research, 5*(4), 481-489. doi:10.1016/j.jare.2013.12.005

Asnar, Y., & Giorgini, P. (2006). Modelling risk and identifying countermeasure in organizations. *Lecture Notes in Computer Science, 4347*, 55-66.

Augusiak, J., Van den Brink, P. J., & Grimm, V. (2014). Merging validation and evaluation of ecological models to 'evaludation': a review of terminology and a practical approach. *Ecological modelling, 280*, 117-128.

Baiardi, F., Suin, S., Telmon, C., & Pioli, M. (2006). Assessing the risk of an information infrastructure through security dependencies. *Critical Information Infrastructures Security*, 42-54.

Bankes, S. (1993). Exploratory Modeling for Policy Analysis. *Operations Research, 41*(3), 435-449. doi:10.1287/opre.41.3.435

Bankes, S., Walker, W. E., & Kwakkel, J. H. (2013). Exploratory Modeling and Analysis. In S. I. Gass & M. C. Fu (Eds.), *Encyclopedia of Operations Research and Management Science* (pp. 532-537). Boston, MA: Springer US.

Berthier, R., Sanders, W. H., & Khurana, H. (2010, 4-6 Oct. 2010). *Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions.* Paper presented at the 2010 First IEEE International Conference on Smart Grid Communications.

Brezhnev, E., Kharchenko, V., Manulik, V., & Leontiev, K. (2018) Critical energy infrastructure safety assurance strategies considering emergent interaction risk. In*: Vol. 582. Advances in Intelligent Systems and Computing* (pp. 67-78).

Brown, G., Carlyle, M., Salmerón, J., & Wood, K. (2006). Defending critical infrastructure. *Interfaces, 36*(6), 530-544.

Buttyan, L., Gessner, D., Hessler, A., & Langendoerfer, P. (2010). Application of wireless sensor networks in critical infrastructure protection: challenges and design options [Security and Privacy in Emerging Wireless Networks]. *IEEE Wireless Communications, 17*(5), 44-49. doi:10.1109/MWC.2010.5601957

Byres, E. (2004). *The Myths and Facts behind Cyber Security Risks for Industrial Control Systems* (Vol. 7).

Cárdenas, A. A., Amin, S., Lin, Z. S., Huang, Y. L., Huang, C. Y., & Sastry, S. (2011). *Attacks against process control systems: Risk assessment, detection, and response.* Paper presented at the Proceedings of the 6th International Symposium on Information, Computer and Communications Security, ASIACCS 2011.

Carnell, R. (2016). lhs: Latin Hypercube Samples. *R package version 0.16, URL: https://cran.r-project.org/web/packages/lhs/index.html*.

Charitoudi, K., & Blyth, A. J. C. (2014). An Agent-Based Socio-Technical Approach to Impact Assessment for Cyber Defense. *Information Security Journal, 23*, 125-136. doi:10.1080/19393555.2014.931492

Clark, R. M., Panguluri, S., Sr., Nelson, T. D., & Wyman, R. P. (2017). Protecting drinking water utilities from cyberthreats. *Journal - American Water Works Association, 109*(2), 50-58. doi:10.5942/jawwa.2017.109.0021

Clarke, R., & Youngstein, T. (2017). Cyberattack on Britain's National Health Service—A Wake-up Call for Modern Medicine. *New England Journal of Medicine, 377*(5), 409-411.

Department of Homeland Security. (2015). *The Future of Smart Cities: Cyber-Physical Infrastructure Risk*. Retrieved from https://ics-cert.us-cert.gov/Future-Smart-Cities-Cyber-Physical-Infrastructure-Risk

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks, 44*(5), 643-666.

Eid, M., & Rosato, V. (2016). Critical Infrastructure Disruption Scenarios Analyses via Simulation. In R. Setola, V. Rosato, E. Kyriakides, & E. Rome (Eds.), *Managing the Complexity of Critical Infrastructures* (pp. 43-61).

Elhamahmy, M., Elmahdy, H. N., & Saroit, I. A. A new approach for evaluating intrusion detection system.

Ericsson, G. N. (2010). Cyber security and power system communication—essential parts of a smart grid infrastructure. *IEEE Transactions on Power Delivery, 25*(3), 1501-1507.

Fairley, P. (2016). Cybersecurity at US utilities due for an upgrade: Tech to detect intrusions into industrial control systems will be mandatory [News]. *IEEE Spectrum, 53*(5), 11-13.

Farwell, J. P., & Rohozinski, R. (2011). Stuxnet and the future of cyber war. *Survival, 53*(1), 23-40.

Formby, D., Durbha, S., & Beyah, R. (2017). Out of control: Ransomware for industrial control systems. In.

Hahn, A., Ashok, A., Sridhar, S., & Govindarasu, M. (2013). Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid. *IEEE Transactions on Smart Grid, 4*(2), 847-855.

Hare, F., & Goldstein, J. (2010). The interdependent security problem in the defense industrial base: An agent-based model on a social network. *International Journal of Critical Infrastructure Protection, 3*(3-4), 128-139. doi:10.1016/j.ijcip.2010.07.001

Herzog, S. (2011). Revisiting the Estonian cyber attacks: Digital threats and multinational responses. *Browser Download This Paper*.

Hull, J., Khurana, H., Markham, T., & Staggs, K. (2012). Staying in control: Cybersecurity and the modern electric grid. *IEEE Power and Energy Magazine, 10*(1), 41-48.

Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences, 80*(5), 973-993. doi:10.1016/j.jcss.2014.02.005

Janssen, S., & Sharpanskykh, A. (2017) Agent-based modelling for security risk assessment. In*: Vol. 10349 LNCS. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 132-143).

Jones, J. (2006). An introduction to factor analysis of information risk (fair). *Norwich Journal of Information Assurance, 2*(1), 67.

Karnouskos, S. (2011). *Stuxnet worm impact on industrial cyber-physical system security.* Paper presented at the IECON Proceedings (Industrial Electronics Conference).

Kaspersky Lab ICS CERT. (2017). *Threat landscape for Industrial Automation Systems in the second half of 2016*. Retrieved from https://ics-cert.kaspersky.com/wp-content/uploads/sites/6/2017/03/KL-ICS-CERT_H2-2016_report_FINAL_EN.pdf

Khurana, H., Hadley, M., Lu, N., & Frincke, D. A. (2010). Smart-grid security issues. *IEEE Security & Privacy, 8*(1).

Korobiichuk, I., Hryshchuk, R., Mamarev, V., Okhrimchuk, V., & Kachniarz, M. (2018) Cyberattack classificator verification. In*: Vol. 635. Advances in Intelligent Systems and Computing* (pp. 402-411).

Kowalski, E., Conway, T., Keverline, S., Williams, M., Cappelli, D., Willke, B., & Moore, A. (2008). *Insider threat study: Illicit cyber activity in the government sector*. Retrieved from

Kruse, C. S., Frederick, B., Jacobson, T., & Monticone, D. K. (2017). Cybersecurity in healthcare: A systematic review of modern threats and trends. *Technology and Health Care, 25*(1), 1-10.

Kwakkel, J. H., & Pruyt, E. (2013). Exploratory Modeling and Analysis, an approach for model-based foresight under deep uncertainty. *Technological Forecasting and Social Change, 80*(3), 419-431. doi:https://doi.org/10.1016/j.techfore.2012.10.005

Lee, R. M., Assante, M. J., & Conway, T. (2016). Analysis of the cyber attack on the Ukrainian power grid. *SANS Industrial Control Systems*.

Li, X., Liang, X., Lu, R., Shen, X., Lin, X., & Zhu, H. (2012). Securing smart grid: cyber attacks, countermeasures, and challenges. *IEEE Communications Magazine, 50*(8), 38-45. doi:10.1109/MCOM.2012.6257525

Liang, G., Weller, S. R., Zhao, J., Luo, F., & Dong, Z. Y. (2017). The 2015 ukraine blackout: Implications for false data injection attacks. *IEEE Transactions on Power Systems, 32*(4), 3317-3318.

Linda, O., Vollmer, T., & Manic, M. (2009, 14-19 June 2009). *Neural Network based Intrusion Detection System for critical infrastructures.* Paper presented at the 2009 International Joint Conference on Neural Networks.

Liu, C. C., Stefanov, A., Hong, J., & Panciatici, P. (2012). Intruders in the grid. *IEEE Power and Energy Magazine, 10*(1), 58-66. doi:10.1109/MPE.2011.943114

Luna, R., Rhine, E., Myhra, M., Sullivan, R., & Kruse, C. S. (2016). Cyber threats to health information systems: a systematic review. *Technology and Health Care, 24*(1), 1-9.

Mansour, N., Chehab, M. I., & Faour, A. (2010). Filtering intrusion detection alarms. *Cluster Computing, 13*(1), 19-29.

Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*: Pearson Education.

May, R. M. (2004). Uses and abuses of mathematics in biology. *Science, 303*(5659), 790-793.

McLaughlin, S., Podkuiko, D., & McDaniel, P. (2009). *Energy theft in the advanced metering infrastructure.* Paper presented at the International Workshop on Critical Information Infrastructures Security.

Miciolino, E. E., Setola, R., Bernieri, G., Panzieri, S., Pascucci, F., & Polycarpou, M. M. (2017). Fault diagnosis and network anomaly detection in water infrastructures. *IEEE Design and Test, 34*(4), 44-51. doi:10.1109/MDAT.2017.2682223

Mikulecky, D. C. (2001). The emergence of complexity: science coming of age or science growing old? *Computers & chemistry, 25*(4), 341-348.

Miller, B., & Rowe, D. (2012). *A survey of SCADA and critical infrastructure incidents.* Paper presented at the Proceedings of the 1st Annual conference on Research in information technology.

Mitchell, R., & Chen, I. R. (2013). Behavior-Rule Based Intrusion Detection Systems for Safety Critical Smart Grid Applications. *IEEE Transactions on Smart Grid, 4*(3), 1254-1263. doi:10.1109/TSG.2013.2258948

Morin, E. (1999). Organization and complexity. *Annals of the New York Academy of Sciences, 879*(1), 115-121.

Moteff, J., Copeland, C., & Fischer, J. (2003). *Critical infrastructures: What makes an infrastructure critical?*

Neuman, C. (2009). *Challenges in security for cyber-physical systems.* Paper presented at the DHS Workshop on Future Directions in Cyber-Physical Systems Security.

Nikolic, I., & Kasmire, J. (2013). Theory. In K. H. Van Dam, I. Nikolic, & Z. Lukszo (Eds.), *Agent-based modelling of socio-technical systems* (Vol. 9, pp. 11-68). Dordrecht: Springer Science & Business Media.

Nikolic, I., Van Dam, K. H., & Kasmire, J. (2013). Practice. In K. H. Van Dam, I. Nikolic, & Z. Lukszo (Eds.), *Agent-based modelling of socio-technical systems* (Vol. 9, pp. 73-137). Dordrecht: Springer Science & Business Media.

Ntalampiras, S. (2015). Detection of integrity attacks in cyber-physical critical infrastructures using ensemble modeling. *IEEE Transactions on Industrial Informatics, 11*(1), 104-111. doi:10.1109/TII.2014.2367322

Pasqualetti, F., Dorfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control, 58*(11), 2715-2729. doi:10.1109/TAC.2013.2266831

Patel, A., Taghavi, M., Bakhtiyari, K., & Júnior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of network and computer applications, 36*(1), 25-41.

Pawlick, J., & Zhu, Q. (2017). Strategic Trust in Cloud-Enabled Cyber-Physical Systems with an Application to Glucose Control. *IEEE Transactions on Information Forensics and Security, 12*(12), 2906-2919. doi:10.1109/TIFS.2017.2725224

Pederson, P., Dudenhoeffer, D., Hartley, S., & Permann, M. (2006). Critical infrastructure interdependency modeling: a survey of US and international research. *Idaho National Laboratory, 25*, 27.

Perakslis, E. D. (2014). Cybersecurity in health care. *N Engl J Med, 371*(5), 395-397.

Priest, B. W., Vuksani, E., Wagner, N., Tello, B., Carter, K. M., & Streilein, W. W. (2015). *Agent-based simulation in support of moving target cyber defense technology development and evaluation.* Paper presented at the Simulation Series.

Puig, V. (2018) Diagnosis and fault-tolerant control of critical infrastructures. In*: Vol. 635. Advances in Intelligent Systems and Computing* (pp. 3-16).

Rasekh, A., Hassanzadeh, A., Mulchandani, S., Modi, S., & Banks, M. K. (2016). Smart Water Networks and Cyber Security. *Journal of Water Resources Planning and Management, 142*(7), 01816004. doi:doi:10.1061/(ASCE)WR.1943-5452.0000646

Rinaldi, S. M. (2004, 5-8 Jan. 2004). *Modeling and simulating critical infrastructures and their interdependencies.* Paper presented at the 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the.

Rinaldi, S. M., Peerenboom, J. P., & Kelly, T. K. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems, 21*(6), 11-25. doi:10.1109/37.969131

Romanosky, S., & Goldman, Z. (2016). Cyber Collateral Damage. *Procedia Computer Science, 95*, 10-17.

Ryan, A. J. (2008). What is a Systems Approach? *arXiv preprint arXiv:0809.1698*.

Rybnicek, M., Tjoa, S., & Poisel, R. (2014) Simulation-based cyber-attack assessment of critical infrastructures. In*: Vol. 191. Lecture Notes in Business Information Processing* (pp. 135-150).

Sandberg, H., Amin, S., & Johansson, K. H. (2015). Cyberphysical security in networked control systems: An introduction to the issue. *IEEE Control Systems, 35*(1), 20-23. doi:10.1109/MCS.2014.2364708

Setola, R., & Theocharidou, M. (2016). Modelling Dependencies Between Critical Infrastructures. In R. Setola, V. Rosato, E. Kyriakides, & E. Rome (Eds.), *Managing the Complexity of Critical Infrastructures* (pp. 19-42).

Shea, D. A. (2004). *Critical infrastructure: Control systems and the terrorist threat*.

Sridhar, S., Hahn, A., & Govindarasu, M. (2012). Cyber–physical system security for the electric power grid. *Proceedings of the IEEE, 100*(1), 210-224.

Stamp, J., Dillinger, J., Young, W., & DePoy, J. (2003). Common vulnerabilities in critical infrastructure control systems. *SAND2003-1772C. Sandia National Laboratories*.

Staniford-Chen, S., Cheung, S., Crawford, R., Dilger, M., Frank, J., Hoagland, J., . . . Zerkle, D. (1996). *GrIDS-a graph based intrusion detection system for large networks.* Paper presented at the Proceedings of the 19th national information systems security conference.

Svendsen, N. K., & Wolthusen, S. D. (2007). Connectivity models of interdependency in mixed-type critical infrastructure networks. *Information Security Technical Report, 12*(1), 44-55. doi:https://doi.org/10.1016/j.istr.2007.02.005

Teixeira, A., Amin, S., Sandberg, H., Johansson, K. H., & Sastry, S. S. (2010). *Cyber security analysis of state estimators in electric power systems.* Paper presented at the 2010 49th IEEE Conference on Decision and Control (CDC).

Ten, C.-W., Liu, C.-C., & Manimaran, G. (2008). Vulnerability assessment of cybersecurity for SCADA systems. *IEEE Transactions on Power Systems, 23*(4), 1836-1846.

Ten, C.-W., Manimaran, G., & Liu, C.-C. (2010). Cybersecurity for critical infrastructures: Attack and defense modeling. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 40*(4), 853-865.

Thompson, B., Morris-King, J., & Harang, R. (2016). *Slowing the spread of Bluetooth-based malware in mobile tactical networks.* Paper presented at the Proceedings - IEEE Military Communications Conference MILCOM.

Tisue, S., & Wilensky, U. (2004). *NetLogo: Design and implementation of a multi-agent modeling environment.* Paper presented at the Proceedings of agent.

Van Dam, K. H., Nikolic, I., & Lukszo, Z. (2012). *Agent-based modelling of socio-technical systems* (Vol. 9): Springer Science & Business Media.

Van den Berg, J., Van Zoggel, J., Snels, M., Van Leeuwen, M., Boeke, S., van de Koppen, L., . . . De Bos, T. (2014). *On (the Emergence of) Cyber Security Science and its Challenges for Cyber Security Education.* Paper presented at the Proceedings of the NATO IST-122 Cyber Security Science and Engineering Symposium.

Van der Lei, T. E., Bekebrede, G., & Nikolic, I. (2010). Critical infrastructures: a review from a complex adaptive systems perspective. *International Journal of Critical Infrastructures, 6*(4), 380-401.

Vasilomanolakis, E., Karuppayah, S., Muhlhauser, M., & Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys, 47*(4). doi:10.1145/2716260

Verizon. (2015). *2015 Data Breach Investigations Report*. Retrieved from https://www.verizonenterprise.com/resources/reports/rp_data-breach-investigation-report_2015_en_xg.pdf

Vuković, O., Sou, K. C., Dán, G., & Sandberg, H. (2012). Network-aware mitigation of data integrity attacks on power system state estimation. *IEEE Journal on Selected Areas in Communications, 30*(6), 1108-1118. doi:10.1109/JSAC.2012.120709

Waldrop, M. (1992). Complexity: The emerging science at the edge of order and chaos. In: New York: Simon&Schuster Paperbacks.

Wei, G., Morris, T., Reaves, B., & Richey, D. (2010, 18-20 Oct. 2010). *On SCADA control system command and response injection and intrusion detection.* Paper presented at the 2010 eCrime Researchers Summit.

Wickham, H., & Chang, W. (2008). ggplot2: An implementation of the Grammar of Graphics. *R package version 0.7, URL: http://CRAN.R-project.org/package=ggplot2*.

Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A survey on cyber security for smart grid communications. *IEEE Communications Surveys and tutorials, 14*(4), 998-1010.

Yuan, Y., Zhu, Q., Sun, F., Wang, Q., & Basar, T. (2013). *Resilient control of cyber-physical systems against Denial-of-Service attacks.* Paper presented at the Proceedings - 2013 6th International Symposium on Resilient Control Systems, ISRCS 2013.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on systems, Man, and Cybernetics*(1), 28-44.

Zhang, Y., Wang, L., Sun, W., Green, R. C., & Alam, M. (2011). Distributed Intrusion Detection System in a Multi-Layer Network Architecture of Smart Grids. *IEEE Transactions on Smart Grid, 2*(4), 796-808. doi:10.1109/TSG.2011.2159818

Zhu, B., Joseph, A., & Sastry, S. (2011). *A taxonomy of cyber attacks on SCADA systems.* Paper presented at the Proceedings - 2011 IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing, iThings/CPSCom 2011.

# Appendix A: Literature review

This appendix contains the literature review that the academic knowledge gap is based on. The increased frequency and impact of cyber-incidents, such as the Stuxnet worm attack on Iranian power plants, sparked a new wave of scientific research pertaining to resilience and security of cyber-physical control systems (Cárdenas et al., 2011; Department of Homeland Security, 2015; Jang-Jaccard & Nepal, 2014; Karnouskos, 2011). A scoping review of contemporary academic literature is conducted in order to establish core academic principles related to cybersecurity of critical infrastructures.

By studying contemporary and well-cited academic publications, a clear pattern emerges, forming the structure of this literature review. Scientific literature generally fits in one of the following categories:

- *Prescriptive* research of protocols and practices in specific cyber-physical systems which require change.
- *Descriptive* research of incidents and associated security components, how these components failed and anomaly detection practices.
- *Methodological* review of shortcomings in establishing coherent, top-down strategy and how to integrate cyber-security by design.

## Prescriptive research

Many publications provide analysis of cyber-incidents, identifying system components that failed, followed by prescribing change to these components. Fairley (2016) is no exception to this, highlighting the severity of the shock that the Stuxnet attack provided to US power grids. Cybersecurity had long been disregarded as an afterthought, whereas Fairley recognises the need for *detecting* and *thwarting* intrusions by design. A similar view is shared by Karnouskos (2011), who relates this need for detection and mitigation to a complicating factor in critical infrastructures: the sheer complexity of networked sensors and components. Karnouskos further specifies Fairley's notion of cybersecurity by design, yet published his findings five years prior – yet Fairley's call for cybersecurity by design is as relevant as it was before. This timespan highlights the lack of pace at which cyber-physical systems change – the shakeup that major incidents cause do not find their way into coherent, tangible cyber-defensive strategies. In an effort to assess future challenges in cybersecurity, Jang-Jaccard and Nepal (2014) distinguish between critical infrastructures and large-scale networks of embedded sensors, a distinction typically not made. They found that emphasising central control systems in isolation might help provide more tangible solutions, but could fail to include interaction between centralised *Supervisory Control And Data Acquisition* (SCADA) systems and a growingly complex environment. In general, there is wide discussion as to which control mechanism performs best, but little attention is paid to interdependencies between infrastructures (Priest et al., 2015; Rinaldi, 2004). The primary take from this is the requirement for more sophisticated control mechanisms than in traditional cybersecurity cases.

A recurring theme in literature is the notion of accessibility of SCADA systems being both a blessing and a curse: while allowing for increased productivity, it also significantly increases the attack surface (Hahn et al., 2013; Karnouskos, 2011; Sandberg et al., 2015). Cárdenas et al. (2011) are proponents of this view on critical infrastructures and supply a widely cited framework for securing critical assets in infrastructures. This framework works well for providing risk assessment and control effectiveness, but fails to address core issues in the design of critical infrastructures. Brezhnev et al. (2018) make use of the accessibility of SCADA systems by proposing duplication and redistribution of important

assets across system components, increasing the required effort to attack. The heterogeneous nature of critical infrastructures components prevents a consistent, easy updating process (Department of Homeland Security, 2015; Karnouskos, 2011). Besides, dependencies and interdependencies between infrastructure nodes indicate that modelling critical infrastructures should always include interconnections (Rinaldi, 2004).

### Descriptive research

Whereas many publications propose specific changes, Neuman (2009) considers this counterproductive, as it results in operators and regulators tacking on inconsistent, unattainable security requirements. The critical consequences of previously non-critical systems are to be described and require proactive integration of security views. More recently, Clark et al. (2017) agree with this notion, describing the impact of lacklustre security requirements, indicating the need for cybersecurity by design. Many authors of prescriptive research might not fully consider the recursive loop they create: heterogeneous, individual systems are to be changed using individual, specific mechanisms. Crucial to the rush to improve security is the disproportionate damage incurred by critical infrastructure failures, something which Romanosky and Goldman (2016) relate to specific evaluation of attack impact. Understanding the concept of collateral damage, as stated in the first sub-section, is crucial to understanding critical infrastructures in general.

Risk assessment mechanisms should thus take the severity of consequences and the complexity of the environment into account. Teixeira et al. (2010) and Liu et al. (2012) assess the impact of policy interventions based on limited knowledge for both attackers and defenders. Puig (2018), on the other hand, assumes security issues to be an optimisation problem and provides the tools and mechanisms on assessing the exact amount of risk. Similarly, Kowalski et al. (2008) relate to insider threats as rational attackers. There seems to be a lack of consensus regarding the rationality of actors. In reality, these viewpoints might not be exclusive, as decisions are made based on an entity's situational awareness. Their situational awareness might be fully rational or limited rationally, based on temporal circumstances (Alcaraz & Lopez, 2013).

### Methodological review

Some of the leading authors on cybersecurity of critical infrastructures identify the complexity of the problem and include all previously discussed elements into extensible frameworks. Teixeira et al. (2010) and Sandberg et al. (2015) propose analysis tools that include control theory, game theory and network optimisation elements that can be applied to any critical infrastructure. Other authors, such as Formby et al. (2017) provide methods to systematically address issues within an isolated type of infrastructure, but the different approaches to securing critical infrastructures are what caused problems to begin with (Cárdenas et al., 2011). Interestingly, Formby et al. (2017) identify industrial control systems as unique due to their cyber-physical nature, which is shared by most contemporary control systems. Individual solutions might fit their associated system well, but security requirements cannot make this distinction as easily. Hahn et al. (2013) provide the basic structure for attack templates, allowing for any infrastructure to be systematically analysed. Modelling the ecosystem of attacks on critical infrastructures requires using accurate yet flexible models for attack and defence scenarios. More sophisticated coordinated attack templates are required to keep up with ever-evolving cyber-threats.

# Appendix B: Model concept formalisation

*Table B-1: Concept formalisation*

| State | Software data structure | Value range |
|---|---|---|
| *Defenders* | | |
| *Node operation* | Float | $\geq 0\ and\ \leq 1$ |
| *Internal operation factor* | Float | $\geq 0\ and\ \leq 1$ |
| *External operation factor* | Float | $\geq 0\ and\ \leq 1$ |
| *Economic impact* | Float | $\geq 0\ and\ \leq 1$ |
| *Physical impact* | Float | $\geq 0\ and\ \leq 1$ |
| *Perceived node operation* | Float | $\geq 0\ and\ \leq 1$ |
| *Perceived internal operation* | Float | $\geq 0\ and\ \leq 1$ |
| *Perceived external operation* | Float | $\geq 0\ and\ \leq 1$ |
| *Current state* | Integer | $0, 1, 2, 3$ |
| *Attackers* | | |
| *Profile* | Integer | $1, 2, 3$ |
| *Knowledge* | String | $"low", "medium", "high"$ |
| *Economic-preference* | Float | $\geq 0\ and\ \leq 1$ |
| *Physical preference* | Float | $\geq 0\ and\ \leq 1$ |
| *Attack capabilities* | Boolean | $true, false$ for each attack |
| *Dependencies* | | |
| *Weighting* | Float | $\geq 0\ and\ \leq 1$ |
| *Current state* | Integer | $0, 1, 2$ |
| *Global states* | | |
| *User traffic frequency* | Float | $\geq 0\ and\ \leq 1$ |
| *User traffic criticality* | Float | $\geq 0\ and\ \leq 1$ |
| *Detection sensitivity* | Float | $\geq 0\ and\ \leq 1$ |
| *Detection specificity* | Float | $\geq 0\ and\ \leq 1$ |
| *Prevention sensitivity* | Float | $\geq 0\ and\ \leq 1$ |
| *Prevention specificity* | Float | $\geq 0\ and\ \leq 1$ |
| *Alleviation threshold* | Float | $\geq 0\ and\ \leq 1$ |
| *Retention threshold* | Float | $\geq 0\ and\ \leq 1$ |
| *Alleviation duration* | Integer | $1, 2, 3, n$ |
| *Retention duration* | Integer | $1, 2, 3, n$ |
| *Worm spread likelihood* | Float | $\geq 0\ and\ \leq 1$ |
| *Attack frequency* | Float | $\geq 0\ and\ \leq 1$ |
| *Attack duration* | Integer | $1, 2, 3, n$ |
| *Attack powers* | Float | $\geq 0\ and\ \leq 1$ for each attack |
| *Total damage sustained* | Float | $\geq 0$ |
| *Physical damage sustained* | Float | $\geq 0$ |
| *Economic damage sustained* | Float | $\geq 0$ |

# Appendix C: Model formalisation flowcharts

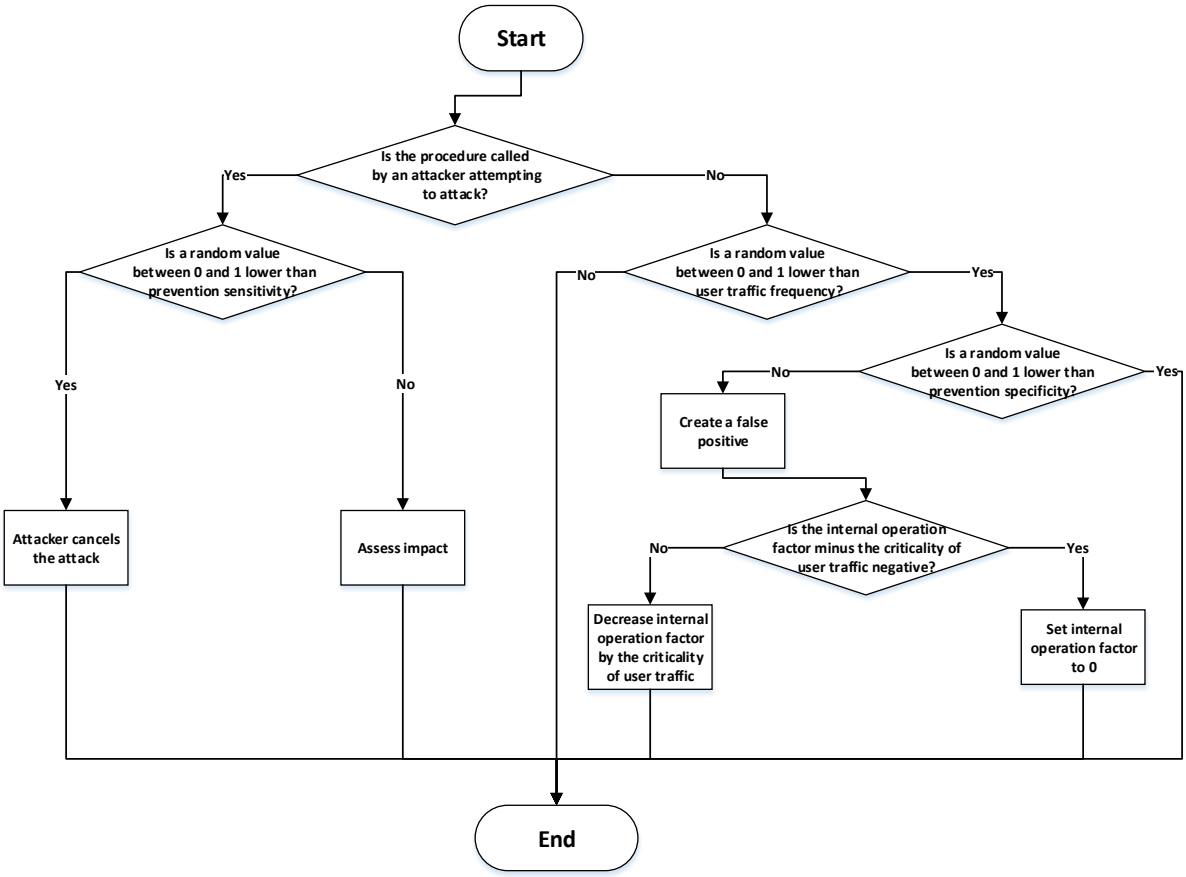


*Figure C-1: Intrusion prevention procedure*

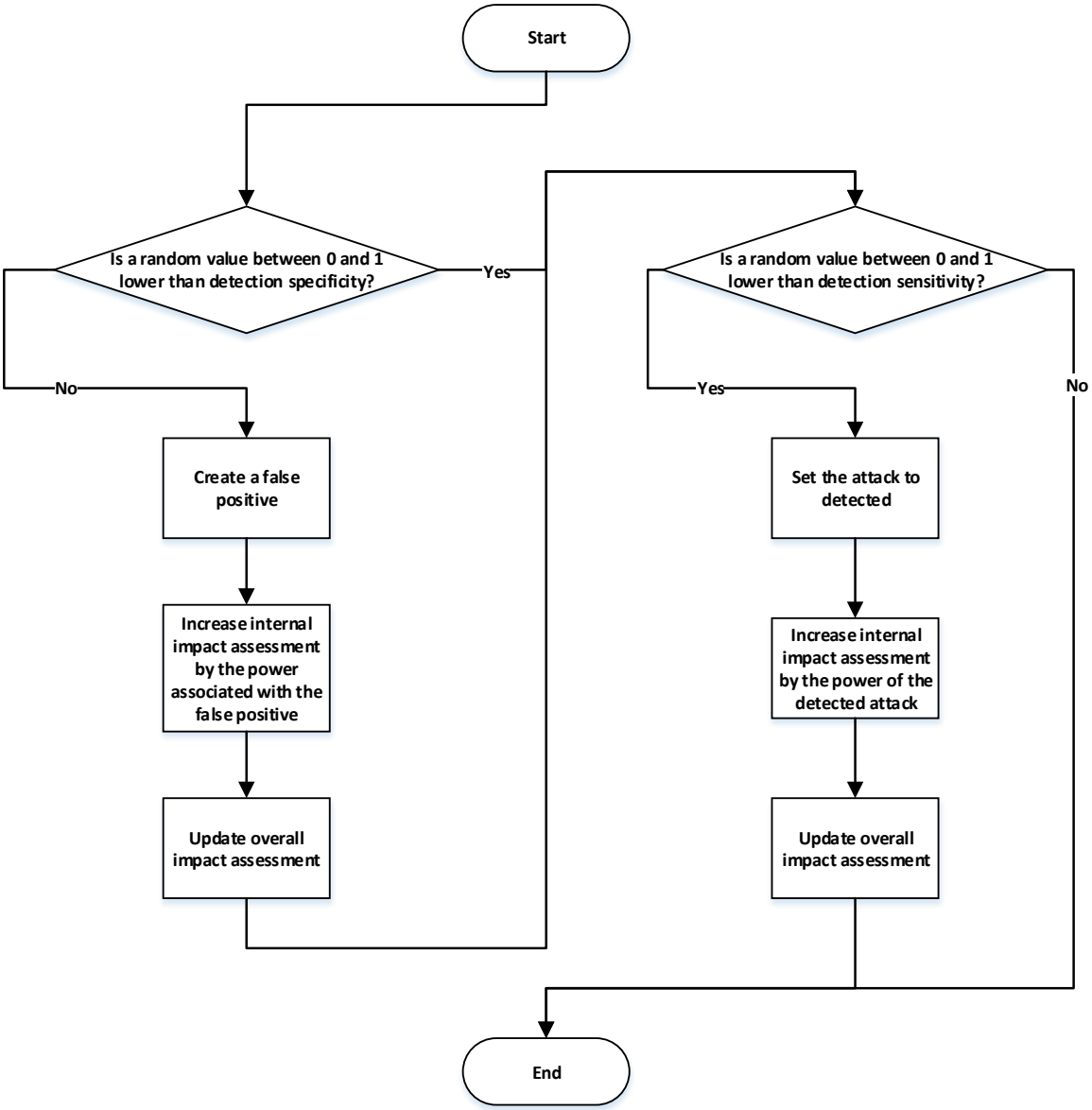


*Figure C-2: Intrusion detection procedure flowchart*

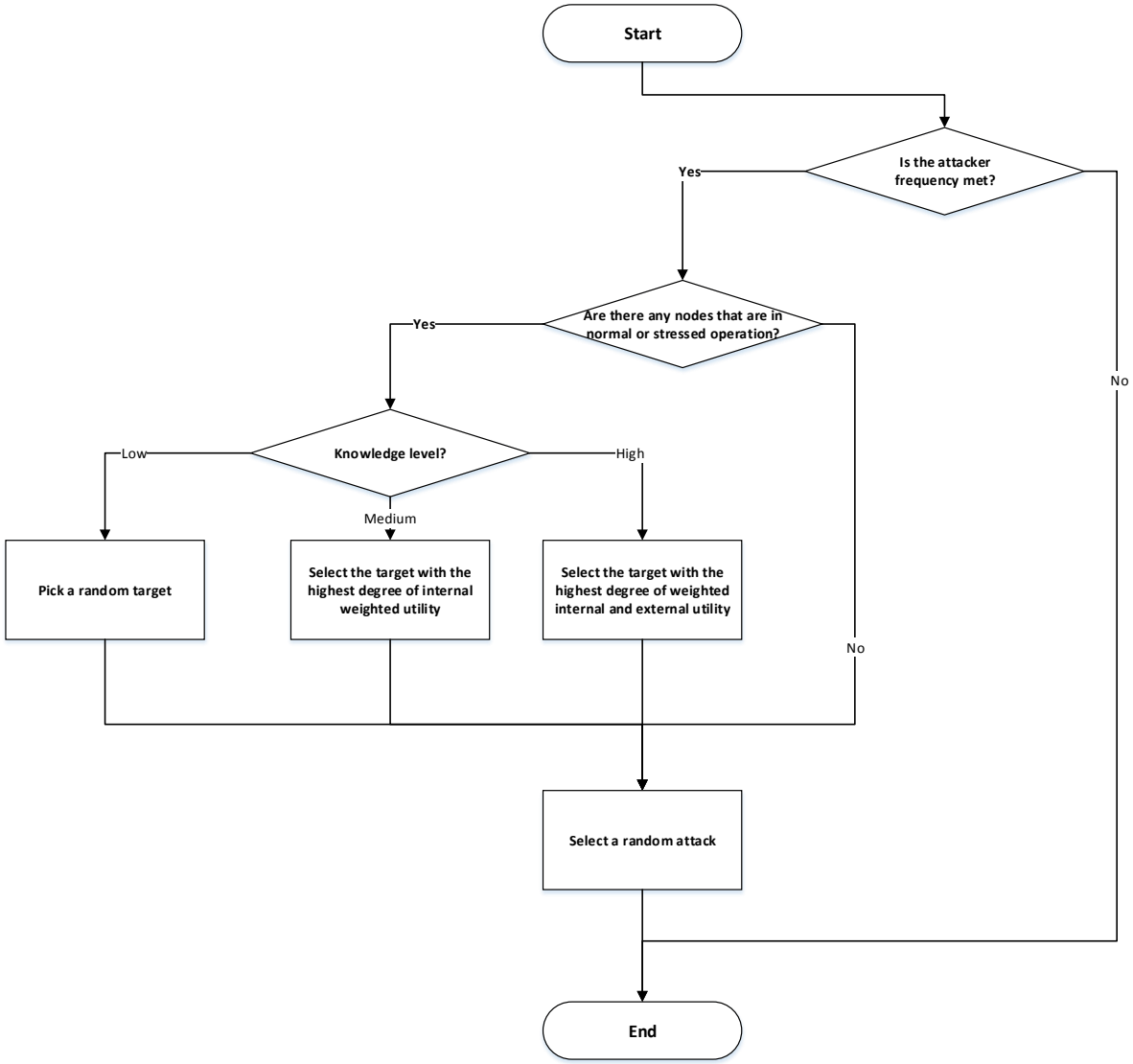


*Figure C-3: Target selection procedure flowchart*

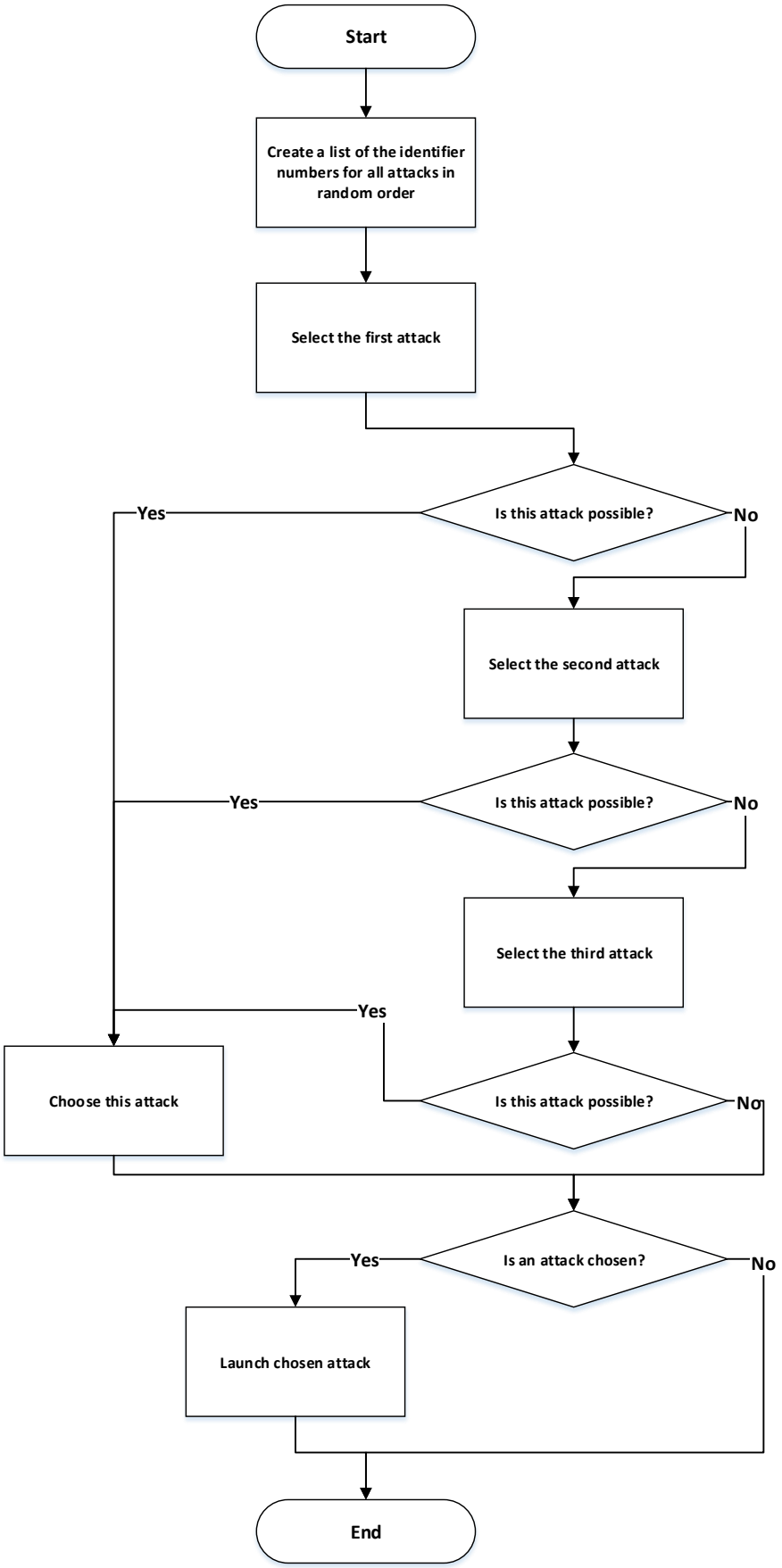


*Figure C-4: Attack selection procedure flowchart*

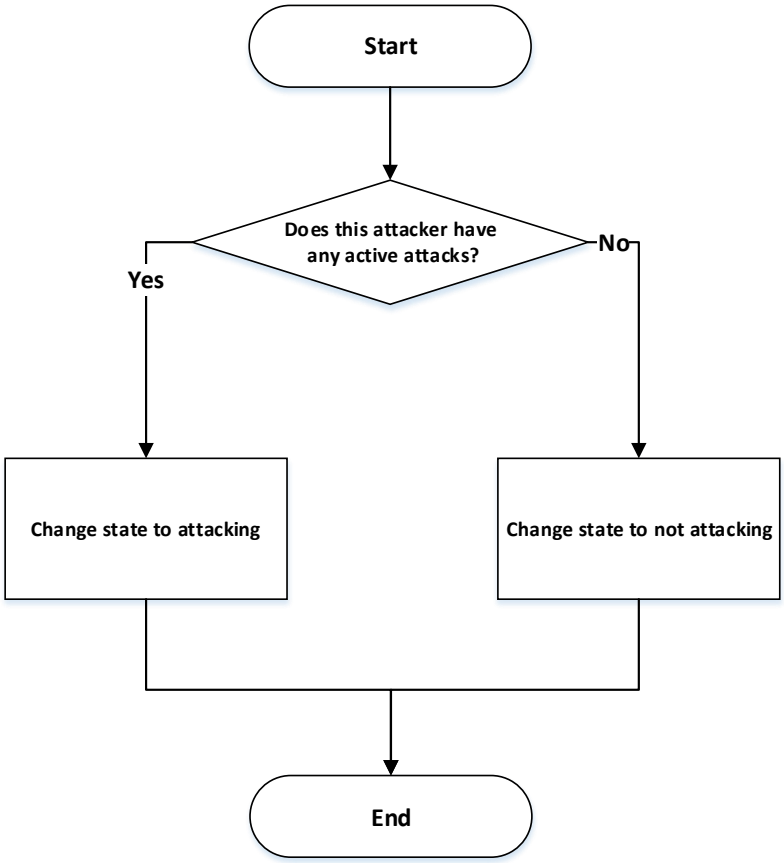


*Figure C-5: Attacker activity assessment procedure flowchart*

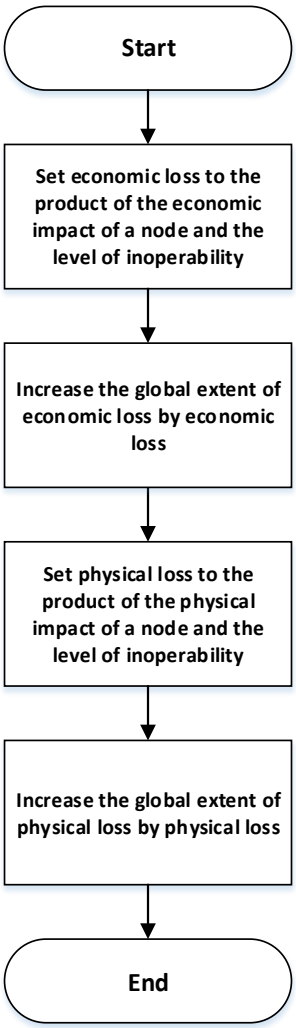


*Figure C-6: Sustain damage procedure flowchart*

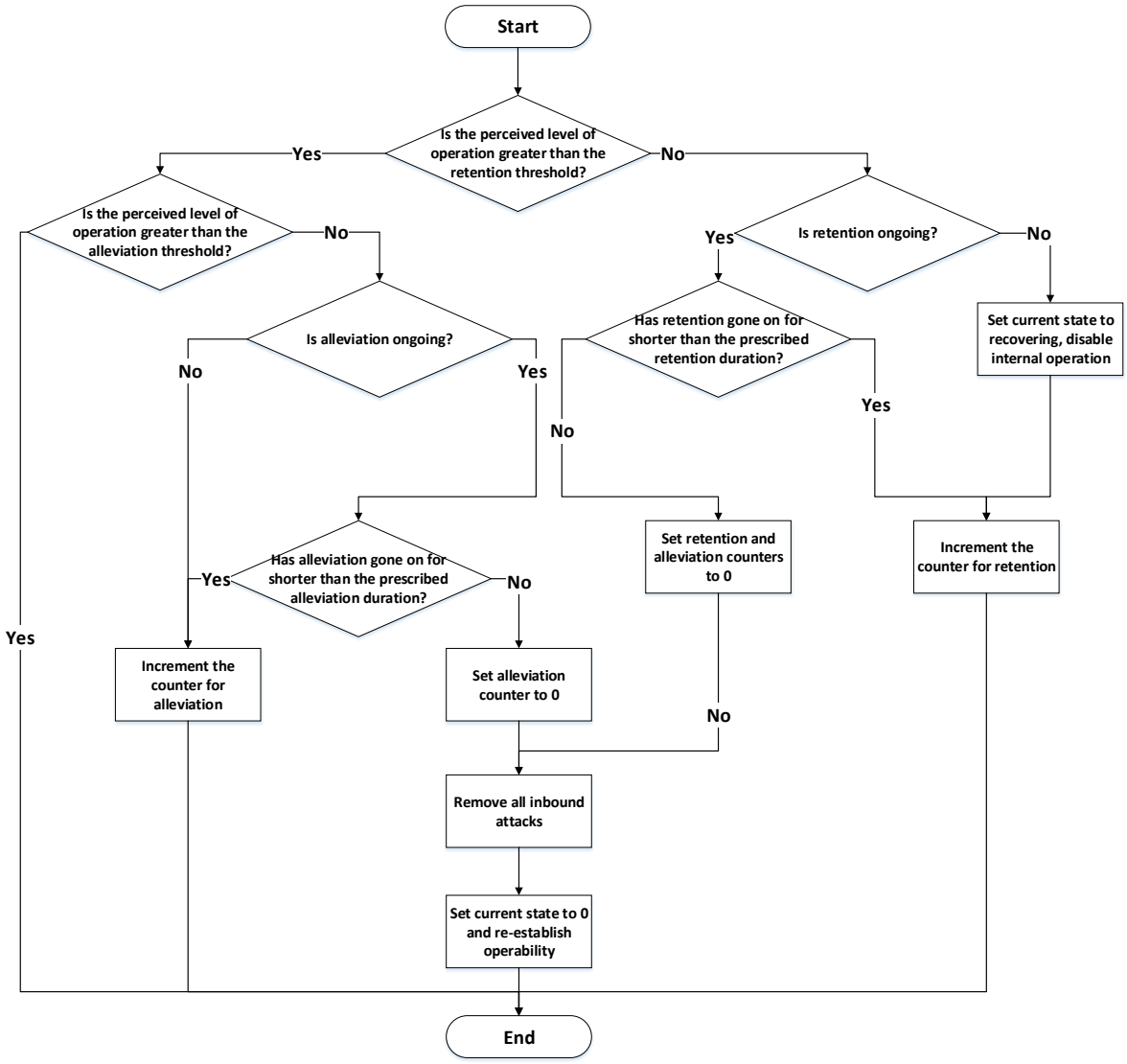


*Figure C-7: Establish response procedure flowchart*

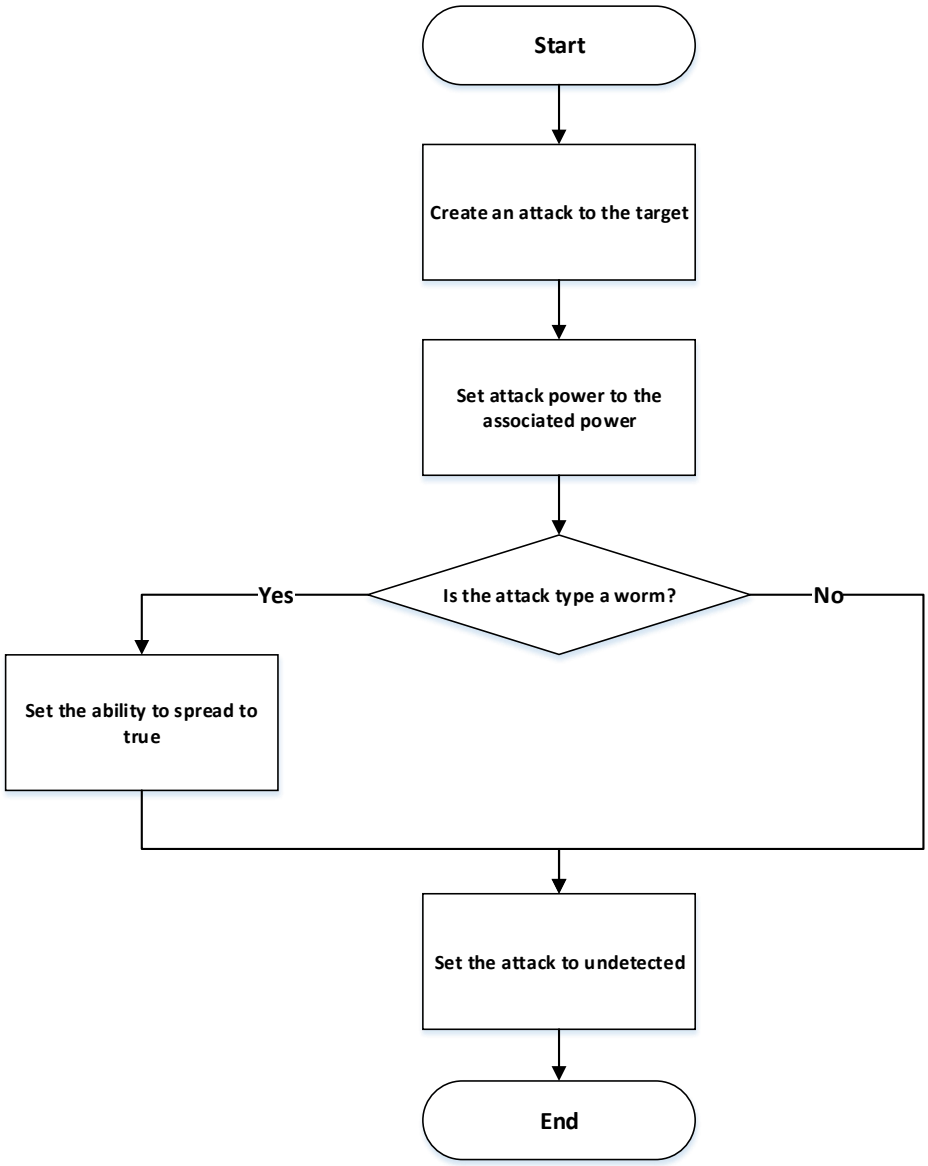


*Figure C-8: Launch attack procedure flowchart*

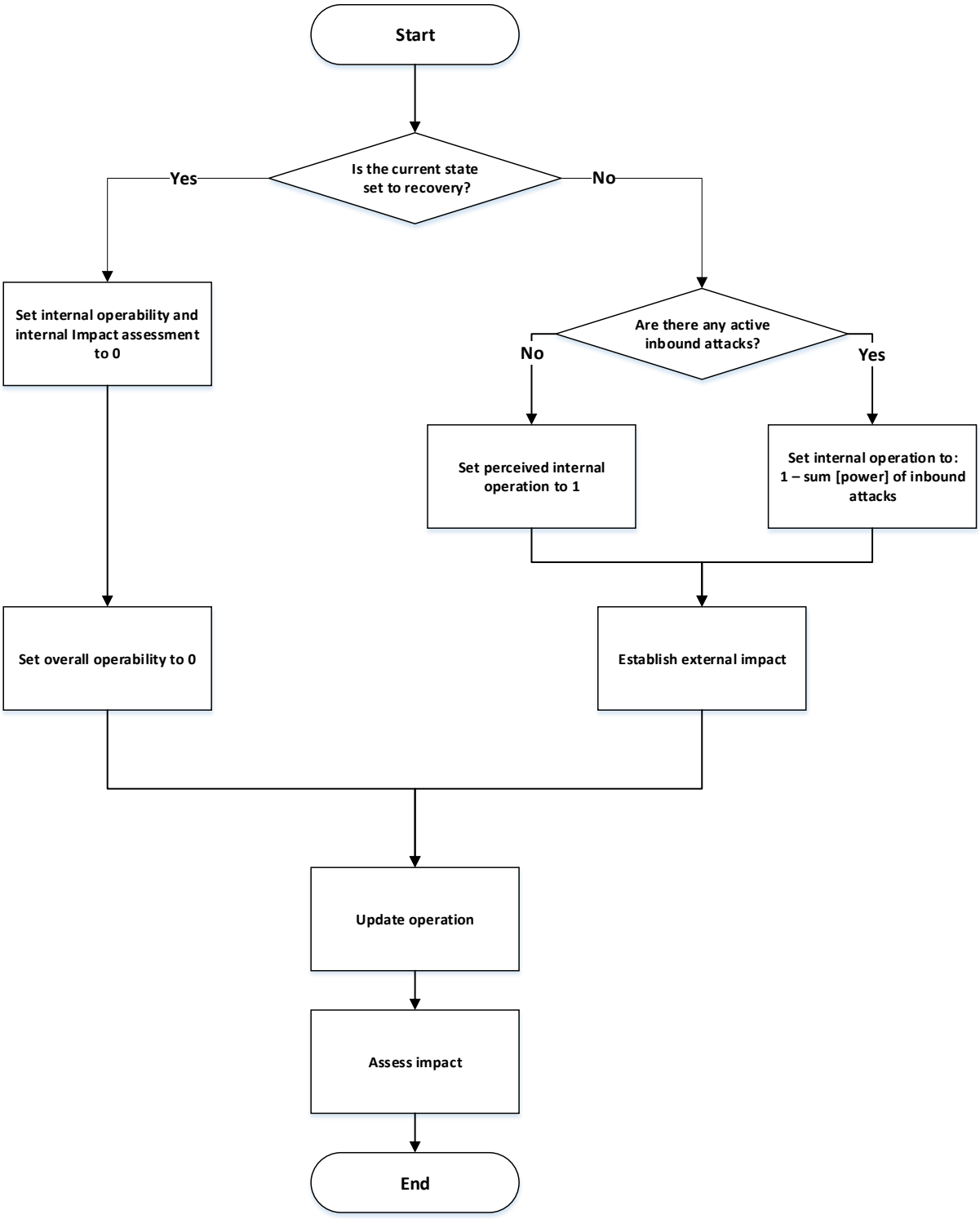


*Figure C-9: Establish operation procedure flowchart*

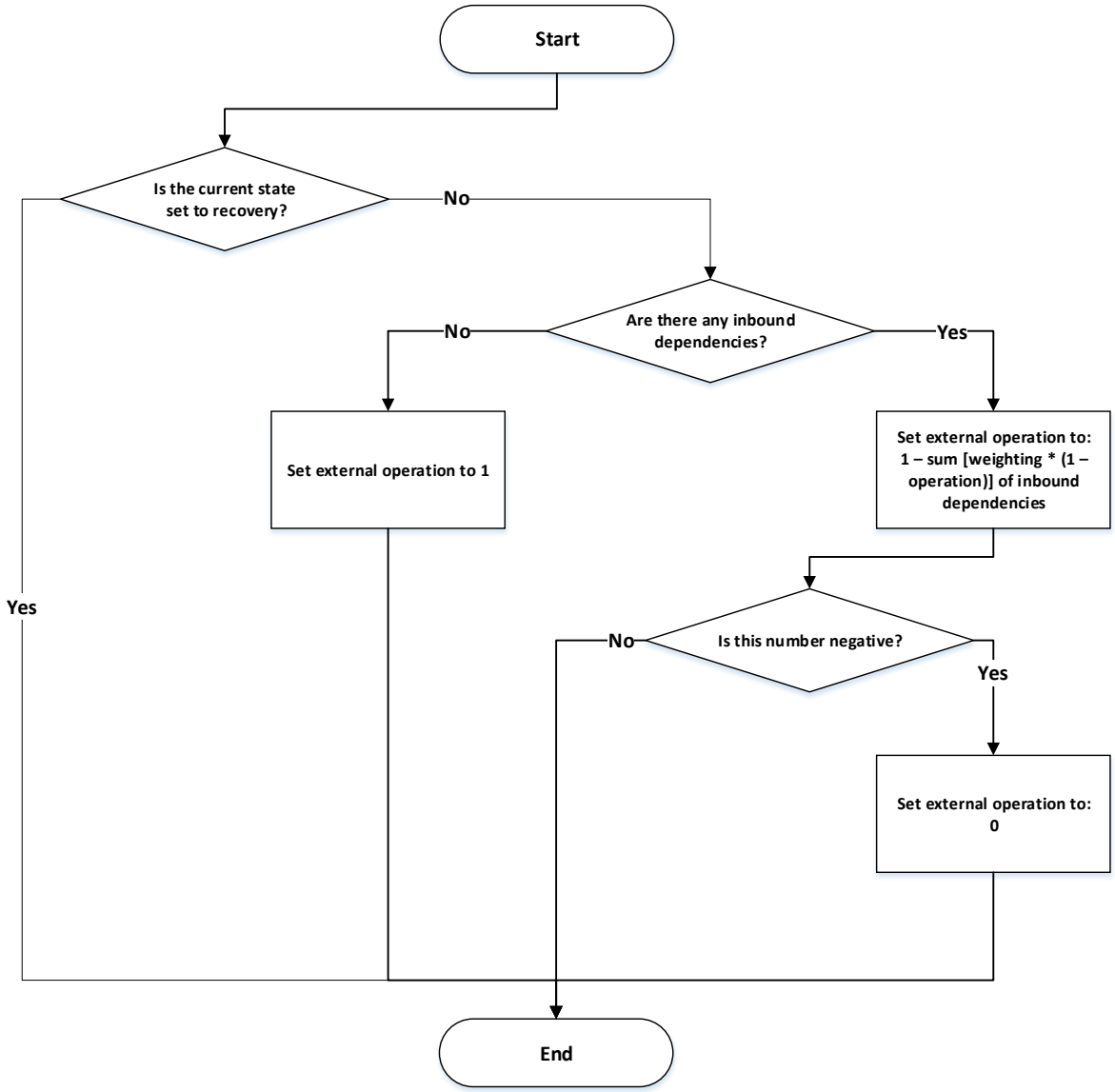


*Figure C-10: Establish external operation procedure flowchart*

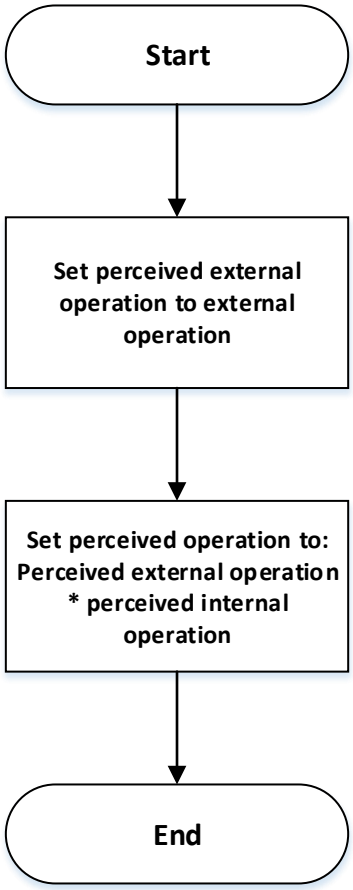


*Figure C-11: Perceive operation procedure flowchart*

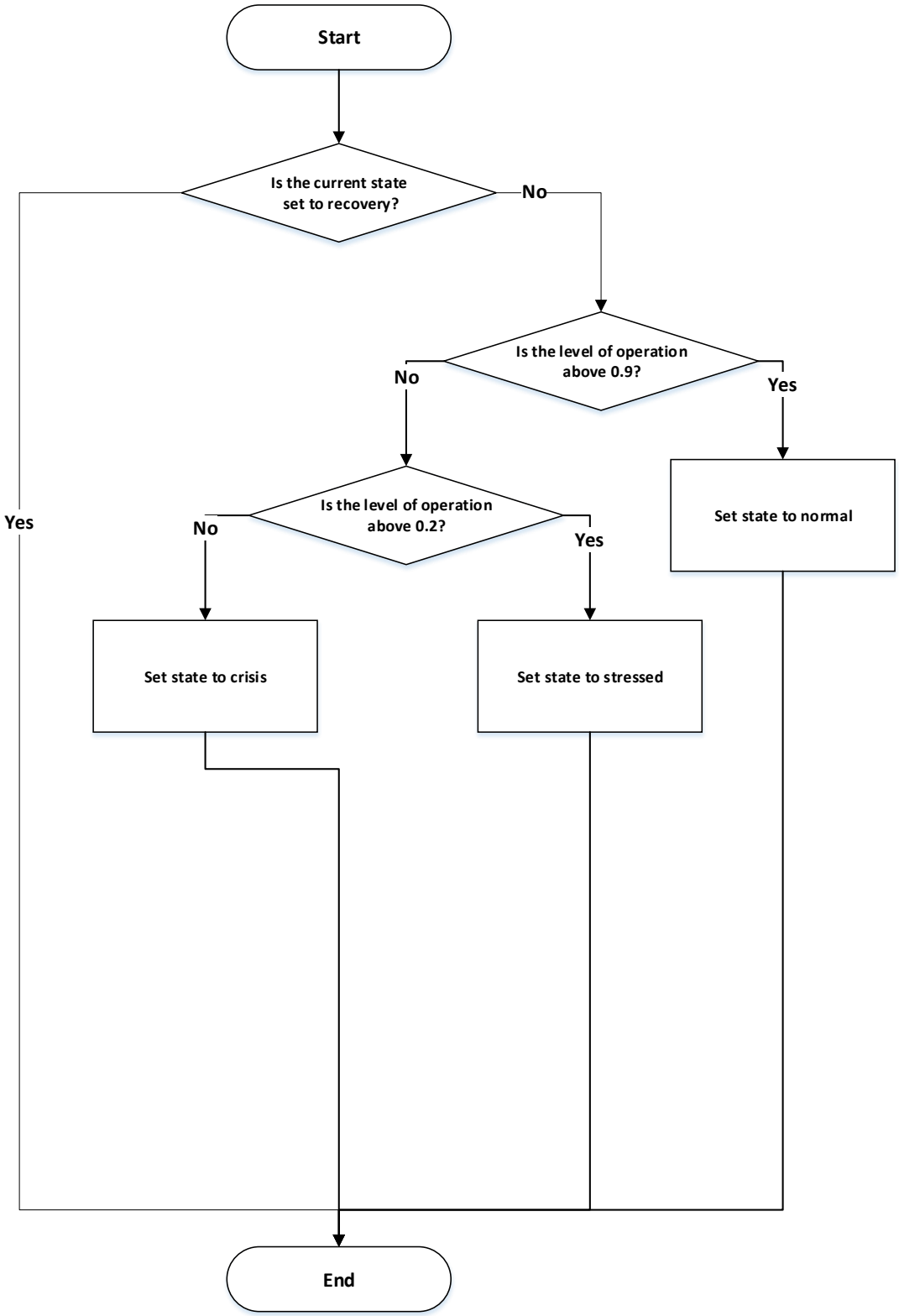


*Figure C-12: Update operation procedure flowchart*

# Appendix D: Modelling assumptions

*Table D-1: List of assumptions adhered to during modelling*

| *Assumption number* | *Description* |
|---|---|
| *Assumption 1* | Attacks do not expire and will remain until they are removed by a defender. |
| *Assumption 2* | Since the model entails an ecosystem, all threats acting against the system do so since they are already motivated. Their decision to attack is motivated by the ecosystem alone and not affected by externally available targets. The prescribed attack frequency alone therefore determines when an attacker initiates an attack. |
| *Assumption 3* | For the same reason as assumption 2, attacks are selected randomly based on the overall population of attacks acting against critical infrastructures. There is no specific preference for any type of attacks other than the distribution of identified types of attacks. |
| *Assumption 4* | All attacks are surmountable by the same procedures of alleviation and retention. During this process, any damage is repaired and consequences are not lasting. |
| *Assumption 5* | The level of operability is represented by a continuous variable that is linearly correlated with the associated losses from node operation. |
| *Assumption 6* | All user traffic shares the same degree of criticality, as further specification of different types of traffic would increase the bias towards specific types of selected infrastructures. |
| *Assumption 7* | False positive classification of legitimate activity is corrected the very next time intrusion detection is applied, as this false positive is not necessarily |
| *Assumption 8* | False positive prevention of legitimate traffic is corrected the next time step, as the crucial instance of traffic is reinitiated |
| *Assumption 9* | If a node is already alleviating intrusions and their impact assessment increases above the threshold for retention, they will instantly change to retaining intrusions. |
| *Assumption 10* | Infrastructure node operators are aware of any inoperability caused by dependencies, even if node operators for the origin node are not aware of inoperability. |
| *Assumption 11* | When a worm spreads to another node, this still counts as normal traffic and will have to bypass intrusion prevention first. |

# Appendix E: Model verification

This appendix denotes the results of verification. Nikolic et al. (2013) prescribe four stages of the verification process, each aiming to establish whether the implemented model procedures match the formalised conceptual model. Each of these steps is followed in the light of the framework provided by Nikolic et al. (2013).

## E.I Tracking agent behaviour

The first step for model verification is to actively track agent behaviour. This was mainly conducted as a continuous process throughout the model implementation phase. Tracking agent behaviour implies using extensive debugging features during the implementation process, which helps identify erroneous coding schemes at an early stage. Problems with the model encountered throughout this phase were immediately dealt with. In terms of verification results, this step is inherently constrained by the complexity of software implementation. As a result, the checks made to ensure agent behaviour matches the formalised model will be discussed for the other verification steps.

## E.II Single agent testing

Single agent testing or unit testing involves testing the behaviour of singular model entities. This is done in order to ensure that a single agent conducts all procedures as formalised and in the right order. A set of checks is formulated to ensure that model behaviour under normal operating inputs is implemented correctly. Furthermore, a set of tests is conducted that seeks to establish possible errors that arise from extreme or impossible values. The latter can help detect problematic model implementation that could possibly obstruct 'normal' model behaviour.

### Test agent sanity: whether computations and choices are made correctly

A list of theoretical predictions was made in order to assess whether computations within the model correspond with the formalised model. Each sanity check is described by the theoretical prediction, the method used to test the prediction and the result of a sanity check.

1. *Theoretical prediction*: the *find-target* procedure is called by any attacker who has no outgoing attacks.
   *Method used: print* command in NetLogo embedded in the procedure for the single agent
   *Result*: **Confirmed**
2. *Theoretical prediction*: an active worm will always call for a chance of spreading
   *Method used*: *print* command in NetLogo embedded in the procedure for the single agent
   *Result*: **Confirmed**
3. *Theoretical prediction*: attacks cannot be launched towards recovering nodes
   *Method used*: *print* command in NetLogo checking whether any target is identified when there are only recovering nodes in the model.
   *Result*: **Confirmed**
4. *Theoretical prediction*: attackers do not launch attacks they are not capable of, even if the expected damage is the highest.
   *Method used: print* commands in NetLogo comparing the types of attack iterated over and comparing these attacks with the chosen type of attack and the list of possible attacks for each attacker profile.
   *Result:* **Confirmed**
5. *Theoretical prediction*: an attacker will randomise their preference for each type of attack and iterate through this list until an attack type is found that the attacker can conduct.
   *Method used*: iterating through model runs with *print* statements embedded in the code. Attackers consistently yielded randomised variants of the list of attack types.

*Result*: **Confirmed**

6. *Theoretical prediction*: If an attacker is not capable of conducting any type of attack, they will still attempt to find a target, but fail to initiate an attack, as they cannot find a viable attack
   *Method used*: setting all attacker switches for attacker/attack combinations to false and running the model. A command was previously implemented in the code to verify whether attacks could be conducted and to provide feedback to model users whether there is an error with parameter settings.
   *Result*: **Confirmed,** the attackers do nothing instead and a warning message is printed in the NetLogo *Command Centre*.
7. *Theoretical prediction*: if all three types of attack have equal power and an attacker is capable of only one type of attack, they will still find the correct identifier associated with that type of attack.
   *Method used*: using *print* statements for the type of attack chosen within the NetLogo code and cross-referencing the chosen attacks with different settings for attacker/attack combinations.
   *Result*: **Confirmed**
8. *Theoretical prediction*: an attacker with a knowledge level of above medium will sort each possible target in descending order based on their expected utility and therefore select the target with the highest degree of utility.
   *Method used:* setting up manual parallel computations of each element involves in utility assessment as the result of *(physical-factor * physical-impact) + (economic-factor * economic-impact)* and comparing whether the chosen node corresponds with the manually assessed node.
   *Result:* **Confirmed**
9. *Theoretical prediction*: If a node is recovering, their perceived internal operation factor and their true internal operation factor are both always set to 0.
   *Method used:* implementing mandatory checks for changing operation factors to ensure nodes are not recovering and verifying these with *print* statements for when any node is recovering and their internal operation factor and perceived internal operation are not equal to 0.
   *Result:* ***Confirmed***
10. *Theoretical prediction*: When the establish-risk procedure is called, this removes the direct effects of previous false positives during intrusion prevention.
    *Method used:* inspecting the node agent between ticks and cross-referencing changes in agent states with *print* statements added to the NetLogo code. Since there are two causes behind changes in true impact, one of them being active attacks, this could easily be verified with the power of active attacks.
    *Result:* **Confirmed**
11. *Theoretical prediction*: A node cannot have its internal operating state reduced below 0 by attacks.
    *Method used*: changing the power of an active attack to a value of greater than 1 and inspecting whether the internal operation factor of the target node is subsequently reduced to 0 or a value below 0.
    *Result*: **Confirmed**
12. *Theoretical prediction*: A node cannot have its internal operating state reduced below 0 by preventing user traffic.

*Method used*: changing user-traffic-criticality to a value greater than 1 and inspecting whether the internal operation factor of a node is subsequently reduced to 0 or a value below 0 when a false positive occurs. This was indicated with a *print* statement.
*Result*: **Confirmed**

13. *Theoretical prediction*: A node cannot have either its perceived operation or its true operation below 0.
*Method used*: changing the power of attack types to values greater than 1 and inspecting whether the perceived internal operation of a target node is subsequently reduced to 0 or a value below 0, while no attack agents were added or removed.
*Result*: **Confirmed**
14. *Theoretical prediction*: A node that meets the operation threshold for alleviation, but not for retention, starts alleviating intrusions. This process goes on even if the operation threshold is no longer met afterwards, until the prescribed duration for alleviation is reached.
*Method used*: setting the perceived internal operation factor for a node to 0.5 while retention-threshold is set to 0.7 and alleviation-threshold is set to 0.3. The node was then manually ordered to conduct the establish-response procedure with *print* statements added to indicate the decision that was made. The perceived internal operation factor was then manually set to 1 (where no threshold would be met) to assess whether alleviation was still ongoing.
*Result*: **Confirmed**
15. *Theoretical prediction*: A node that meets the operation threshold for retention starts retaining intrusions. This process goes on even if the operation threshold is no longer met afterwards, until the prescribed duration for retention is reached. The retention process should also start if alleviation is ongoing, reducing the time needed to fully retain intrusions depending on how long the node had been alleviating intrusions.
*Method used*: setting the perceived internal operation factor for a node to 0.2 while alleviation-threshold is set to 0.3. The node was then manually ordered to conduct the establish-response procedure with *print* statements added to indicate the decision that was made. The perceived internal operation factor was then manually set to 1 (where no threshold would be met) to assess whether retention was still ongoing. The same process was applied to a node that was set to alleviate (a value for alleviation greater than 0).
*Result*: a slight **error** was found in the code that made the required time for retention 1 tick longer if the node had swapped from alleviating to retaining intrusions. The increment for a node's *retention* was moved down in the code, making sure that all iterations of this procedure encounter the increment process. This was then **revalidated** and **confirmed.**
16. *Theoretical prediction*: When a node is confronted with an attack and the node fails to prevent this attack, their internal operation factor is decreased by the power of an attack up to a minimum of 0, and the node updates the true impact subsequently.
*Method used*: setting up a node that was attacked while detection-sensitivity was set to 0 (i.e. no attacks would be prevented). The attack power was set to 0.35. The expectation would then be that the internal operation-factor was set to 0.65. Updating the extent of inflicted damage happens at the end of each tick, after all required interaction has taken place.
*Result*: **Confirmed**
17. *Theoretical prediction*: A node does not incorporate undetected attacks in their impact assessment.
*Method used*: setting up a node that was attacked by two separate attacks, one detected with power 0.35 and another undetected with power 0.55. The expectation would be that

the internal operation factor would be set to 1 – 0.55 – 0.35 = 0.2, and that perceived internal operation would be set to 0.65.
*Result*: **Confirmed**

### Breaking the agent

Besides conducting sanity checks, another useful method for single agent verification is attempting to 'break' agents. By deliberately entering values that exceed the predefined range for several parameters, or by adding extreme values, model behaviour can be uncovered that should be prevented for the sake of robustness. Each test will be described in terms of objectives, results and whether any changes were added to counteract the phenomena.

1. *Agent-breaking action*: changing the power associated with attacks to negative values.
   *Results*: attacks inflict no damage to nodes, as instead the level of operation and perceived operation increases far beyond the range of values prescribed for the parameter. The result is that no attacks are removed, and worms eventually spread across the entire ecosystem of connected nodes. With each attack, the level of perceived operation increases leading to no defensive decisions being made.
   *Remedy*: the only way to achieve this level of operability is by intentionally modifying the values of attacks beyond what the sliders allow. Still, in order to prevent unintended model behaviour, checks can be added each time the true and perceived levels of internal operability are derived to ensure that these values do not extend beyond 1. A check was added to prevent values for perceived-internal-operation and internal-operation-factor from extending beyond 1.
2. *Agent-breaking action*: changing the power associated with attacks to values above 1.
   *Results*: a single attack manages to completely disrupt internal operation of a node. As a result, false positives detected through intrusion detection bring the perceived operability down to 0 instantly. Because checks were already in place to ensure the levels of operability cannot drop below 0, the model behaviour witnessed is in line with expectations.
   *Remedy*: no remedy required, as the checks already in place to prevent negative operability levels behave as expected.
3. *Agent-breaking action*: changing the values for retention duration and alleviation duration in such a way that retention takes longer than alleviation. Retention-duration was set to 14, while alleviation-duration was set to 7.
   *Results*: the model catches this combination of setup parameters and automatically adjusts the value of retention-duration to match that of alleviation-duration. A warning is printed in the NetLogo *Command Center* (*Model setup warning: Retention-duration should never be longer than alleviation-duration. Retention-duration has been set to 7*). Further model behaviour is not changed.
   *Remedy*: no remedy required, as the check for setup parameters caught the logically inconsistent settings.
4. *Agent-breaking action*: changing the values for retention threshold and alleviation threshold in such a way that retention would be conducted before alleviation would be considered. Retention-threshold was set to 0.8, whereas alleviation-threshold was set to 0.7.
   *Results*: the model catches this combination of setup parameters and automatically adjusts the value of retention-threshold to be lower than that of alleviation-threshold, and a warning message in the NetLogo *Command Center* similar to the message of test 3. However, there was an error in this mechanism, as it still resulted in a higher retention-threshold than intended. What should have been a subtraction operator was an addition operator. After fixing this, the correct values were set and the correct message was displayed: *(Model setup*

*warning: Retention-threshold should never be longer than alleviation-threshold. Retention-threshold has been set to 0.69)*.
*Remedy*: As stated under results, the + operator was changed to a − operator, fixing issues encountered and making the pre-emptive check function as intended.

5. *Agent-breaking action*: changing the number of nodes to 1.
   *Results*: the model crashes during setup, as all nodes are requested to create connections and dependencies to other nodes. The model is built around simulating a network of nodes and did not account for the possibility to create only one node.
   *Remedy*: while not strictly necessary, as simulating one node is not representative for any real-world situation within this ecosystem, a single check for creating connections and dependencies was added. These functions are now only called if there are multiple nodes in the model.
6. *Agent-breaking action*: changing intrusion detection and prevention sensitivity and specificity values to negative values.
   *Results*: changing prevention-sensitivity to a negative value results in every attack attempt being successful. There is no difference between negative values and a value of 0, as the chance of succeeding is 0. Changing detection-sensitivity to a negative value works similarly for the detection process, as no attacks are detected at all. Negative values for prevention-specificity lead to most nodes being constantly stressed, as all user traffic results in false positives. Negative values for detection-specificity lead to nodes constantly detecting non-existent attacks and near-ubiquitous decisions to alleviate or retain intrusions, even if there are no attacks.
   *Remedy*: no remedy required, the model functions as expected.
7. *Agent-breaking action*: changing utility preference parameters for attackers to negative values.
   *Results*: attackers still select the target based on the highest value for perceived utility, however this value is now negative. If an attacker has values for physical-preference of -0.3 and economic-preference of -0.7, the attacker would prefer targets with relatively high physical-impact. All values are negatives, but are still sorted in descending order with the highest value of perceived utility being selected.
   *Remedy*: no remedy required, the model functions as expected.
8. *Agent-breaking action*: changing attacker/attack combinations to make all attacks impossible across the model.
   *Results*: as described during sanity check 6 for single-agent testing, attackers are forced to verify whether they can conduct each attack in order of associated power. If an attacker cannot select an attack, they will abort the procedure and a message is relayed to the model user in the NetLogo *Command Center*: *((attacker 34) detected incorrect model parameter usage. Could not find any attack to conduct, aborting procedure.).*
   *Remedy*: no remedy required, the model functions as expected.

## E.III Interaction testing in a minimal model

Besides paying attention to a single agent to verify whether their actions correspond with the formalised model, another approach is to test interaction in a minimal model. This implies selecting the bare minimum of agents required for all basic interaction. In the case of this model, this results in one attacker agent and two node agents. Two node agents are required to incorporate connection links and dependency links, which fulfil a crucial role in interaction. Each test for interaction is described along a scenario for which multiple predictions are made for interaction taking place and the values this results in.

1. There is an attacker with economic-preference of 1 and physical-preference of 0.5, in the following scenario:

   *There are two connected and interdependent nodes (A and B), with values for respectively economic impact and physical impact of 0.75 and 0.25 for A and 0.25 and 0.75 for B, and the weighting of the dependency from A->B is 0.3 whereas the weighting of dependency B->A is 0.7*

   The attacker would make the following decisions:
   - An attacker with "low" knowledge will randomly select either
     *Method used:* running multiple iterations and seeing whether any variety is encountered that could be directly attributed to the random seed used.
     *Result*: **Confirmed**
   - An attacker with "medium" knowledge will select node A with perceived utility of 0.875 over node B with perceived utility of 0.625
     *Method used:* implementing *print* statements that verify whether the expected value matches the encountered value for perceived utility.
     - A: $1 \times 0.75 + 0.5 \times 0.25 = 0.875$
     - B: $1 \times 0.25 + 0.5 \times 0.75 = 0.625$

     *Result*: **Confirmed**
   - An attacker with "high" knowledge will select node B with perceived utility of 1.2375 over node A with perceived utility of 1.0625
     *Method used:* implementing *print* statements that verify whether the expected value matches the encountered value for perceived utility.
     - A: $0.875 + 0.3 \times 0.625 = 1.0625$
     - B: $0.625 + 0.7 \times 0.875 = 1.2375$
     - *Note: Due to internal data references, NetLogo yielded the value 1.23749999 repeating for node B, which is internally interpreted as 1.2375. The internal model implementation can therefore be confirmed as verified.*

     *Result:* **Confirmed**
2. Given the following scenario:

   *There are two connected and interdependent nodes (A and B), with dependency weightings of 0.8 from A->B and 0.5 from B->A. The alleviation-threshold and retention-threshold are set to 0.7 and 0.3, respectively. For the sake of reproducibility, the lone attacker is only capable of conducting one type of attack, a worm with associated power 0.5.*

   If the attacker decides to attack node A, the following chain of events is expected:
   - *Event:* the attacker attempts to create an attack to node A with associated power 0.5 and worm? set to true. The creation of the attack is subject to intrusion prevention: if this is successful, nothing happens and the chain stops here. If intrusion prevention was unsuccessful, subsequent steps are encountered.
     *Observation:* in the first instance, the attacker failed to pass the intrusion prevention check. The attack was successfully prevented and no further action was warranted. In the second instance, intrusion prevention was unsuccessful and an attack was created with associated power 0.5 and worm? set to true.
     *Result:* **Confirmed**
   - *Event:* Following the creation of the attack, node A adjusts the internal operation level to $1 - 0.5 = 0.5$.
     *Method used:* internal-operation-factor was correctly reduced to 0.5 as a result of the attack.
     *Result:* **Confirmed**

- *Event:* Node A changes current-state accordingly in line with the new level of operation and changes the outgoing dependency state. Both states should be set to 1.
  *Observation:* node A changed its state to correspond with the level of operation and shifted to represent the yellow-toned *stressed* state, codified as current-state = 1.
  *Result:* **Confirmed**
- *Event:* Node B updates their true impact and impact assessment by given the inbound dependency, which affects the level of operation at node B by $0.5 \times 0.8$. The new level of operation should therefore be 0.6. Node B should change their state to 1, as a result, and so should the dependency from B->A. Node A then reacts to this shift in external perturbation and adapts its overall level of operation.
  *Observation:* node B detected the change in state for the dependency and shifted its external-operation factor to $1 - 0.8 \times (1 - 0.5) = 0.6$. Node A is now affected by the dependency from B->A and adapts its external-operation-factor to $1 - 0.5 \times (1 - 0.6) = 0.8$. The internal level of operation in node A is now $0.8 \times 0.5 = 0.4$.
  *Result:* **Confirmed**
  *Note:* all of these events took place on one tick. Two interdependent nodes form a closed loop which will continuously decrease the level of external operability in both nodes that react to each other. The best way to unambiguously verify the values that take place without treading into precise numbers with more a vast number of decimals is to assess whether updates are correctly computed within the one tick assessed.
- *Event:* Node A detects the attack and updates their perceived-internal-operation to 0.5 and subsequently changes the value for perceived-operation to $0.5 \times 0.8 = 0.4$.
  *Observation:* the moment the attack on node A becomes detected, node A shifts their perceived-internal-operation from 1 to 0.5. As a result, node A computes perceived-operation to $0.5 \times 0.8 = 0.4$.
  *Result:* **Confirmed**
- *Event:* Node A decides on the appropriate response and chooses alleviation, as the retention-threshold is not met, whereas the alleviation-threshold is met. The value for alleviation should be increased to 1, and a label with the letter *A* should appear on the node.
  *Observation:* node A detects the attack, which turns blue, and updates their perceived-internal-operation to $1 - 0.5 = 0.5$. Based on this, node A starts the alleviation procedure and increments the value for alleviation, as well as displaying an *A*.
  *Result:* **Confirmed**
- *Event:* Attack spreads to node B, which is undetected. Node B updates internal-operation-factor and operation based on the true impact of the attack.
  *Observation:* after the attack spreads to node B, the internal-operation-factor for node B is reduced by 1 to 0.5, similarly to what happened to node A before.
  *Result:* **Confirmed**

## E.IV Multi-agent testing

The fourth and last step of model verification is to conduct multi-agent testing. Multi-agent testing involves verifying model behaviour for the entire model with default parameterisation, as shown in Appendix F. This implies that agents are present in normal quantities. The aim for this step is to identify whether behavioural patterns that emerge are consistent with behaviour that matches

hypothesised or desired behavioural patterns in the real world. There are two main approaches to conducting multi-agent testing: variability testing and timeline sanity analysis. Both of these steps will be conducted and related to model behaviour for each key performance indicator discussed.

## Variability testing

Variability testing involves running many repetitions of the same model, typically 100 to 1000, to suppress heavily chaotic behaviour and help normalise emergent patterns (Nikolic et al., 2013). The aim of variability testing is to hypothesise and discuss whether the degree of variability across performance indicators is explainable. Timeline sanity testing involves simulating a low number of instances of the same model to assess whether behaviour can be explained by other parameters. The setup used for these verification steps are 1000 repetitions used to assess the variability of each parameter and only the first three repetitions used to discuss timeline sanity. Each of these variability tests are conducted using boxplot graphs over the set of 1000 repetitions, grouping together sets of 10 model ticks. This shows the general tendencies for data to exert itself between boundaries given by the upper and lower quartiles, as well as showing the behaviour of the majority of data points in the second and third quartiles.

### *Damage to nodes*

The first set of performance indicators discussed are those indicators relating to the extent of damage to nodes, discussed in section 5.4. The first indicator assessed is the total extent of losses incurred by nodes over time**.** Losses should gradually increase over time as they are caused by any slight disruption in nodes. Cumulative losses can only increase over time, determined by the extent by which operability is hampered. This is inherently rooted in run-specific circumstances and should show variability among repetitions. The variability across cumulative losses indicate the extent to which losses are incurred and shows the relative difference among pure system performance.

The associated boxplot is shown in Figure E-1. As can be seen in the graph, the majority of data points trend along the same near-linear relationship shown by the second and third quartiles. This shows a steady increase in the extent of losses over time, which is as expected. Variations in the growth of losses in several outliers is expected, yet the differences between outliers and general means are rather slim. Upper outliers can be explained due to more sensitive iterations of node configurations, as in some cases nodes with a high degree of perceived utility might also carry heavier dependencies. This highlights the need for a high number of repetitions for experimentation in order to catch the overall possible model outcomes. This is corroborated by the extent of current or contemporary losses at each point in time, shown in Figure E-2. This plot shows how the linear coefficient for cumulative losses trends between around 0.10 and 0.15 per node for the majority of repetitions. This implies the average degree of operability is between 0.85 and 0.90 for these repetitions. The other elements of the boxplot graph show that there is variability possible, but as expected the behaviour this exerts does not change over time. The reactive nature of the simulation model is identified in the comparability of each time step. Outliers at each point in time are similarly the result of coincidental configurations of nodes and dependencies.

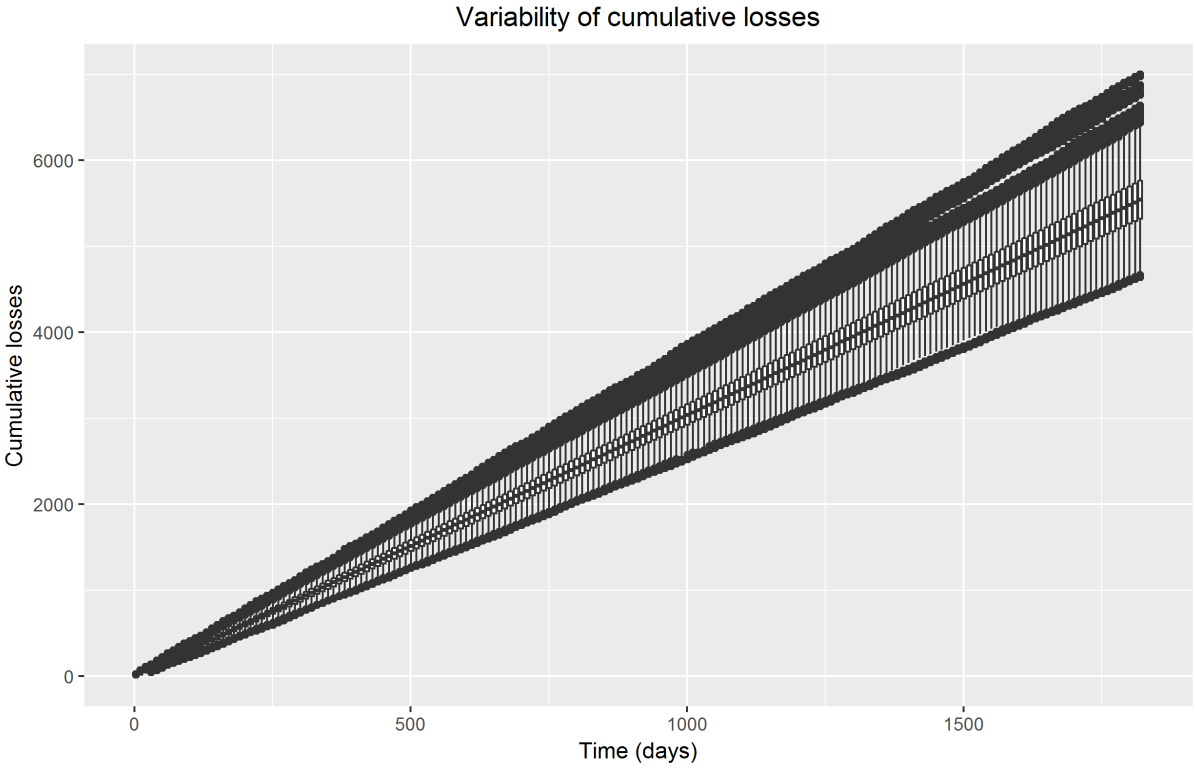


*Figure E-1: Cumulative losses over time for variability testing*

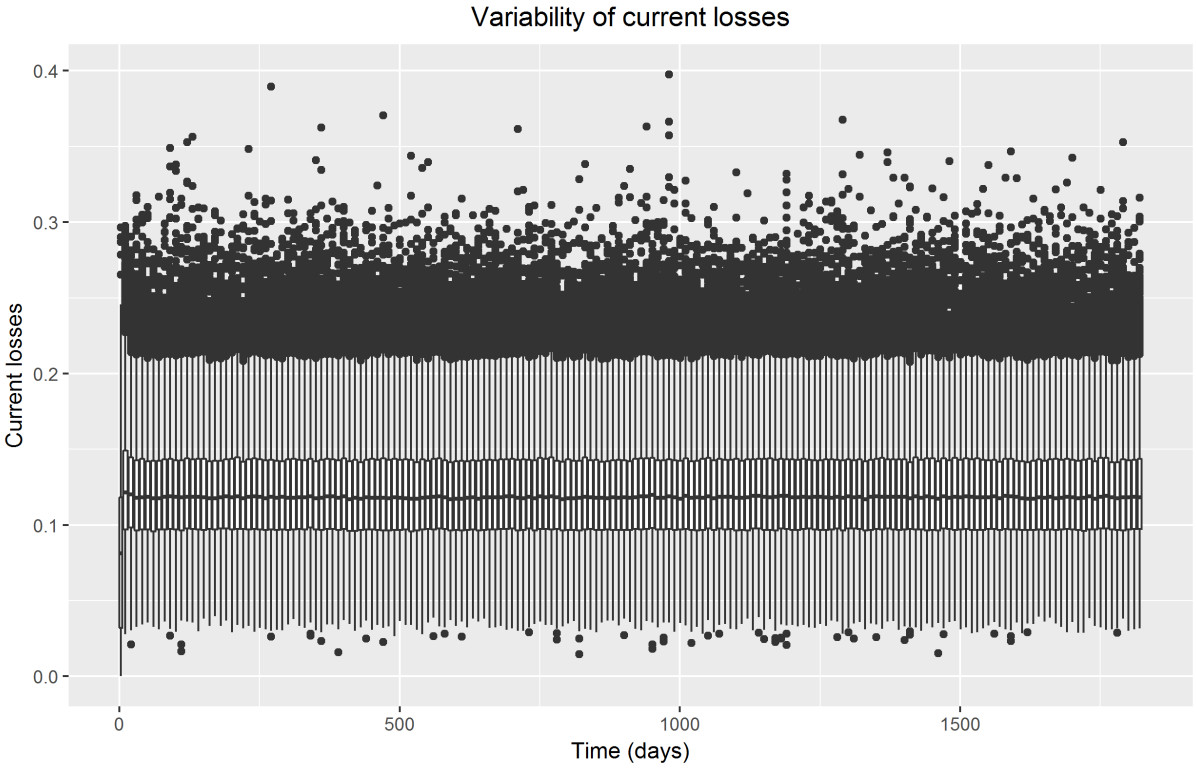


*Figure E-2: Current losses per node over time for variability testing*

The second indicator assessed is the development of node operation states over time. This gives a brief indication of the level of operability that can be attained within the simulation run. Since each run implicitly incorporates a degree of chaos, the expectation is that this parameter will show a great

degree of variability across each time step, given many different repetitions are inspected. Given the foundations of the formalised model, emphasis is placed on the effectiveness of defensive strategies within the ecosystem, which in itself implies a degree of reactiveness. The implicit degree of reactiveness means that entities within the model are supposed to react to contemporary events. On the basis of analysing 1000 repetitions of the simulation model, the mean distribution of node operation status should be comparable across each time step. This was identified for the first indicator, as it directly relates to the extent of losses sustained. The upper and lower limits and standard deviations across model runs are supposed to show variety between each time step, but will ultimately trend within similar boundaries. The expectations for these plots is, as with the previous indicator, that there is little variability between time intervals. However, it is expected that the number of *normal* and *stressed* nodes interchange to a certain extent. These two plots should indicate similar spread. On the other hand, *inoperable* nodes should occur much less and cases of multiple inoperable nodes should be rare. The plots for the numbers of *normal*, *stressed* and *inoperable* nodes are respectively shown in Figure E-3, Figure E-4 and Figure E-5.

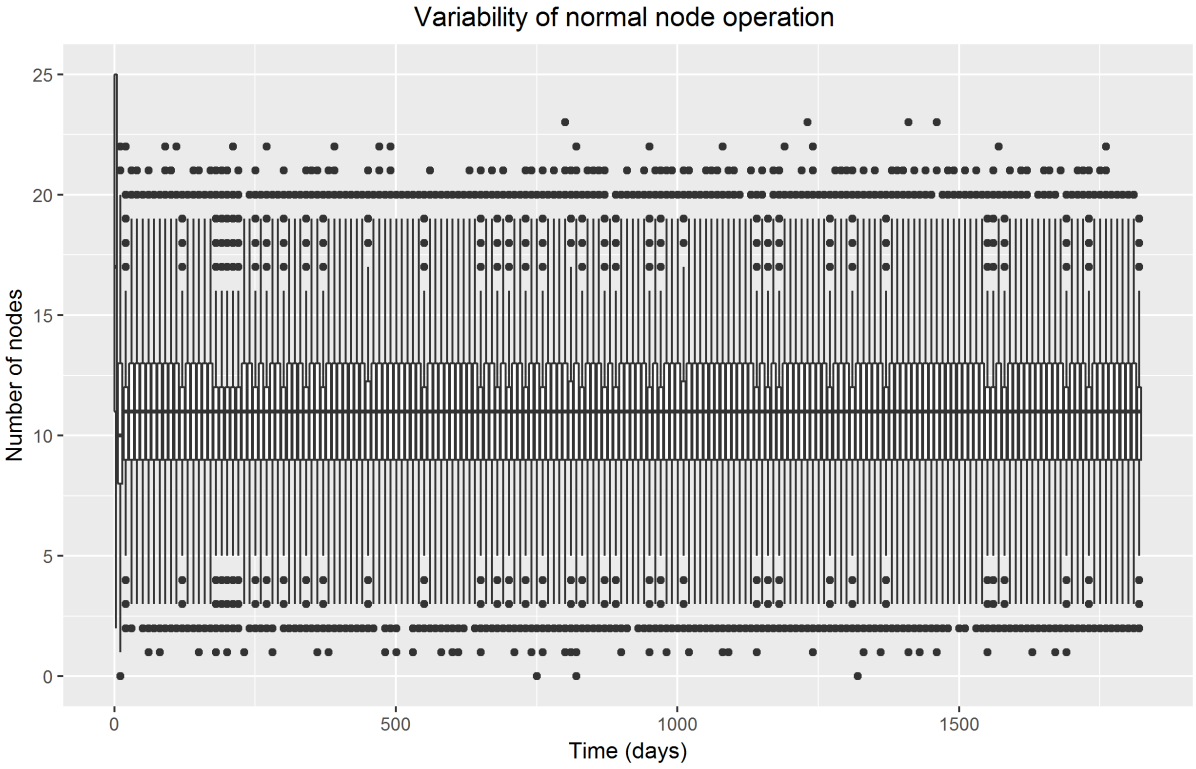


*Figure E-3: Number of normal nodes over time for variability testing*

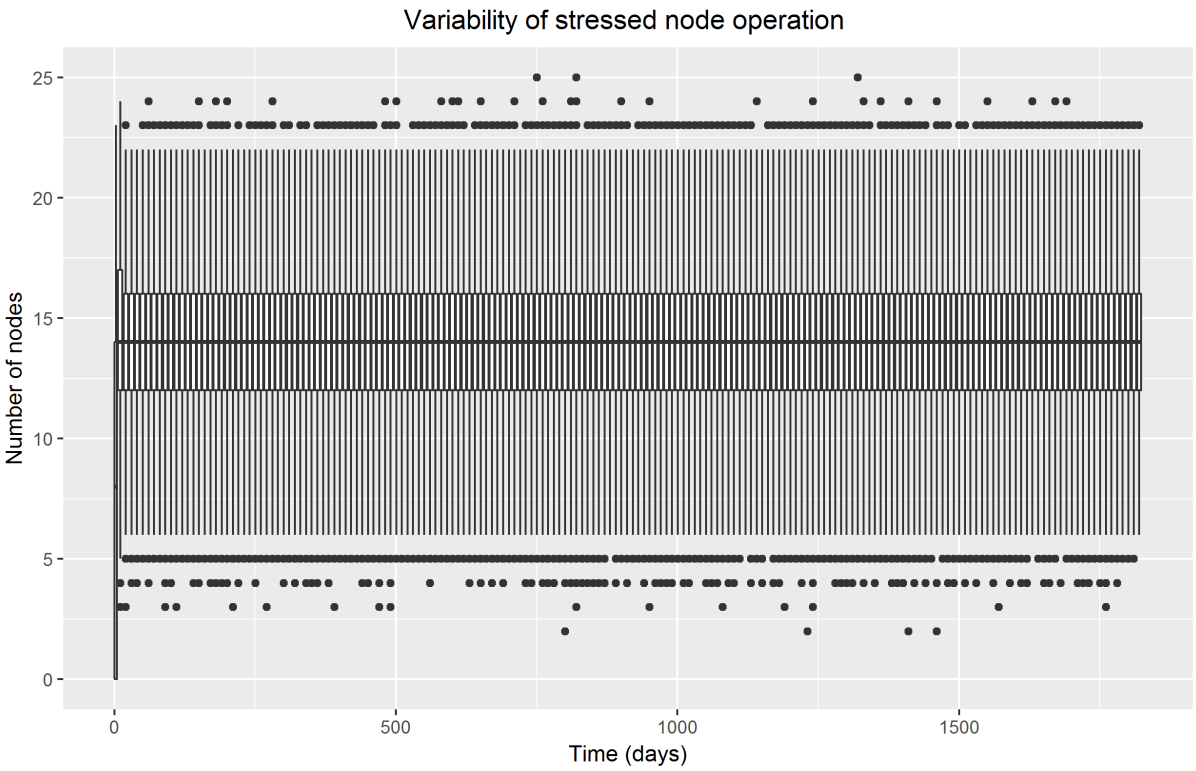


*Figure E-4: Number of stressed nodes over time for variability testing*

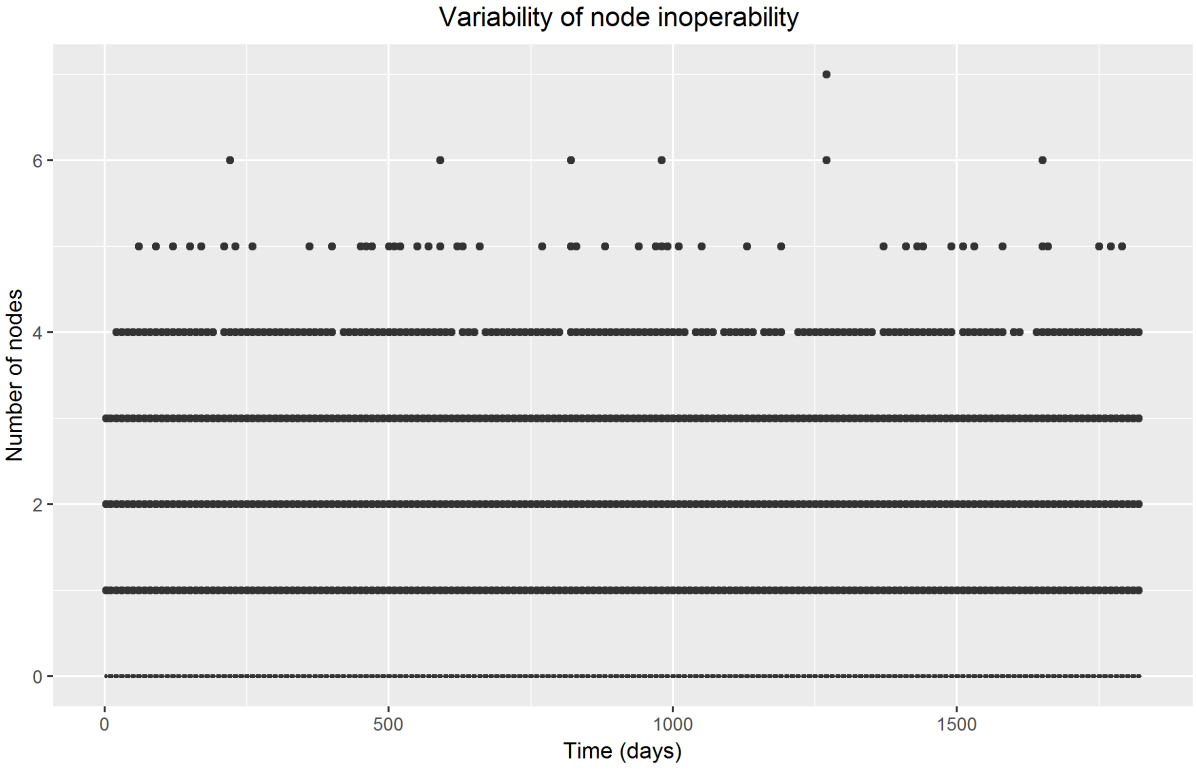


*Figure E-5: Number of inoperable nodes over time for variability testing*

The plots show significant variation across the number of normal and stressed nodes, whereas the number of inoperable nodes trends heavily at 0. It should be noted that the scales are different for inoperable nodes to highlight the rarity of these phenomena. Out of the 1825000 total observations

(1000 repetitions of 1825 ticks), only around 14.8% show an inoperable node count above 0. This emergent behaviour replicates the desired behaviour, where slight disruptions in infrastructure operation lead to a slight increase in losses and the frequency of inoperable infrastructure nodes is low. Not seen in the plots, the number of normal and stressed nodes shows significant variation across model runs: single run model behaviour is not a static data point, as one might deduce from these plots. This will be further detailed during timeline sanity testing.

*Quality of defensive decisions*

The second set of performance indicators discussed relate to the overall quality of defensive decisions made across a model run. The first indicator used is the deviation between true operability of nodes and the perceived level of operation of nodes over the course of each type of decision made. The expectation is, once again, little variability between time intervals across the 1000 simulation runs. Furthermore, significantly high or significantly low values would indicate there are implementation errors for impact assessment. This highlights the extent by which impact assessment yielded incorrect information at each time step. The associated plot is shown in Figure E-6. As can be seen from the boxplot, the majority of repetitions show little variance, with almost perfect symmetry across the mean over all time steps. One thing that should be noted is the presence of outliers on the lower boundaries of the plot. Many values are identified which slightly exceed the fourth quartile, yet would not be sufficient to be incorporated in this fourth quartile. The fact that these outliers occur on the lower end more so than the higher end of the deviation in impact assessment likely implies that the set of parameters used for these repetitions tends to underestimate the level of operation on a node.

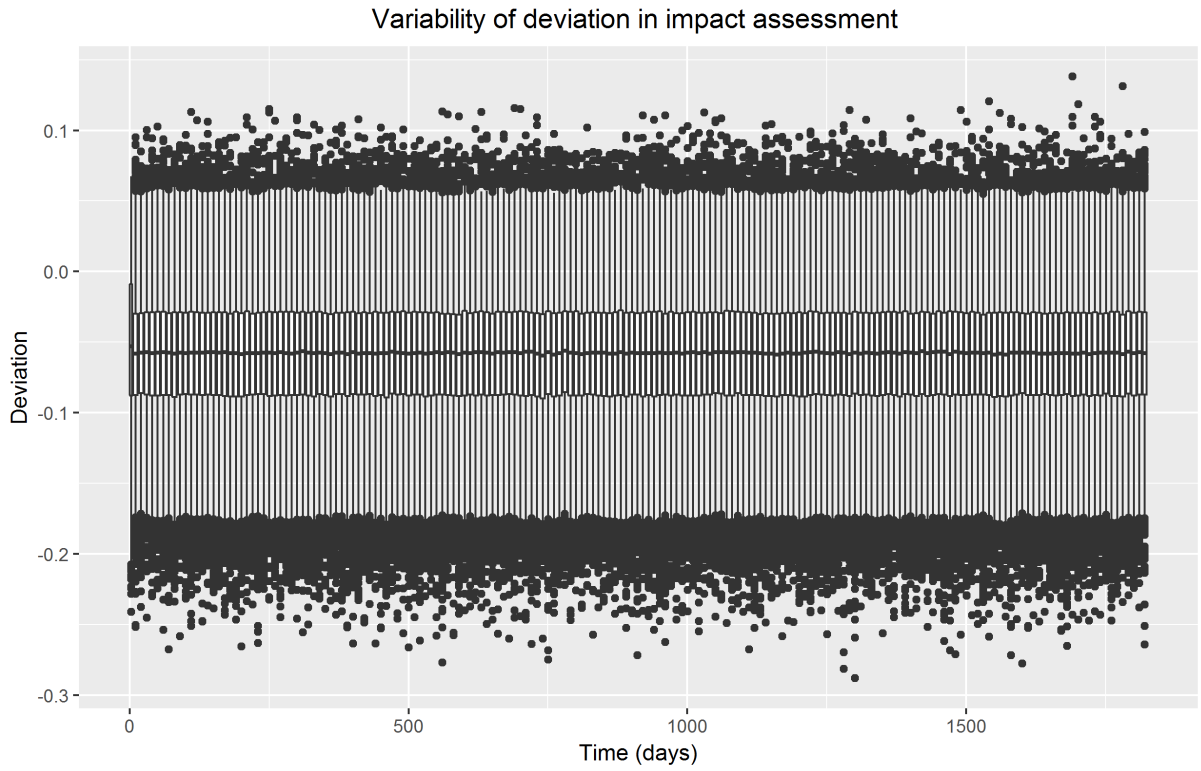


*Figure E-6: Average deviation in impact assessment over time for variability testing*

The second indicator assessed is the frequency at which each type of defensive decision (do nothing, alleviate or retain), plots for which are shown in Figure E-7Figure E-7: Frequency of decisions made as part of the total number of decisions for variability testing. Figure E-7a and Figure E-7b show a similar pattern as seen before, as a near-symmetrical distribution of data points is shown across all model

runs. The second and third quartiles of each boxplot are densely clustered around the mean and the upper and lower quartiles cover similar areas on each respective side of the mean. For Figure E-7c, all boxplots cover the value 0, as most observations do not include any decisions to retain intrusions. All observations where these do occur are therefore inherently considered outliers for boxplot graphs. There appears to be no substantial change in behaviour or distribution across different time steps, as expected. The total possible value range is substantial when compared to the mean, but it is not obvious from this graph alone whether these differences originate from chaotic model behaviour or stable differences between separate model repetitions. Considering timeline sanity analysis described further down and differences observed through descriptive statistics, which will be expanded upon further, it can be concluded that model runs result in only minor overall differences: the observed symmetry is the result of similar behaviour at different time steps across model runs, showing no unexpected variability.

The third performance indicator assessed relates to the correctness or quality of defensive decisions made. The variability of these parameters across the set of 1000 repetitions is shown in Figure E-8. As with the frequency of defensive decisions, the observed distribution of data points is stable across different time steps and shows nearly symmetrical behaviour. Figure E-8a and Figure E-8c together show that the majority of decisions made tends to be either correct or based on an underestimation of operability given the parameter settings used for variability testing. Figure E-8b shows that occasionally, decisions are made based on overestimating operability, but these tend to be outliers. In terms of expected variability, these performance indicators are inherently likely to change over time within a single model repetition. The level of symmetry indicates that no implementation errors can be deducted from these 1000 repetitions, as model behaviour follows expected chaos between time steps within individual repetitions.

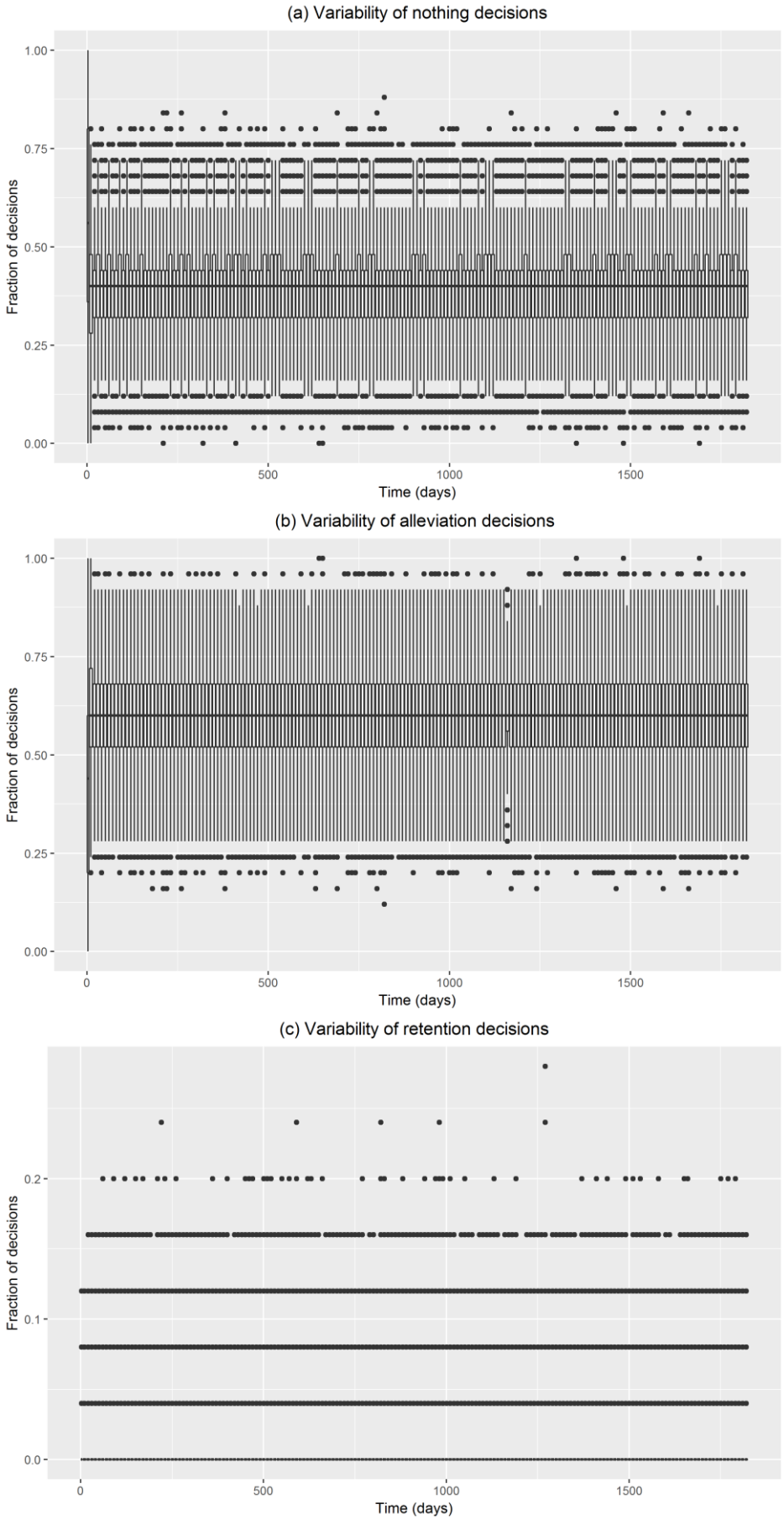


*Figure E-7: Frequency of decisions made as part of the total number of decisions for variability testing*

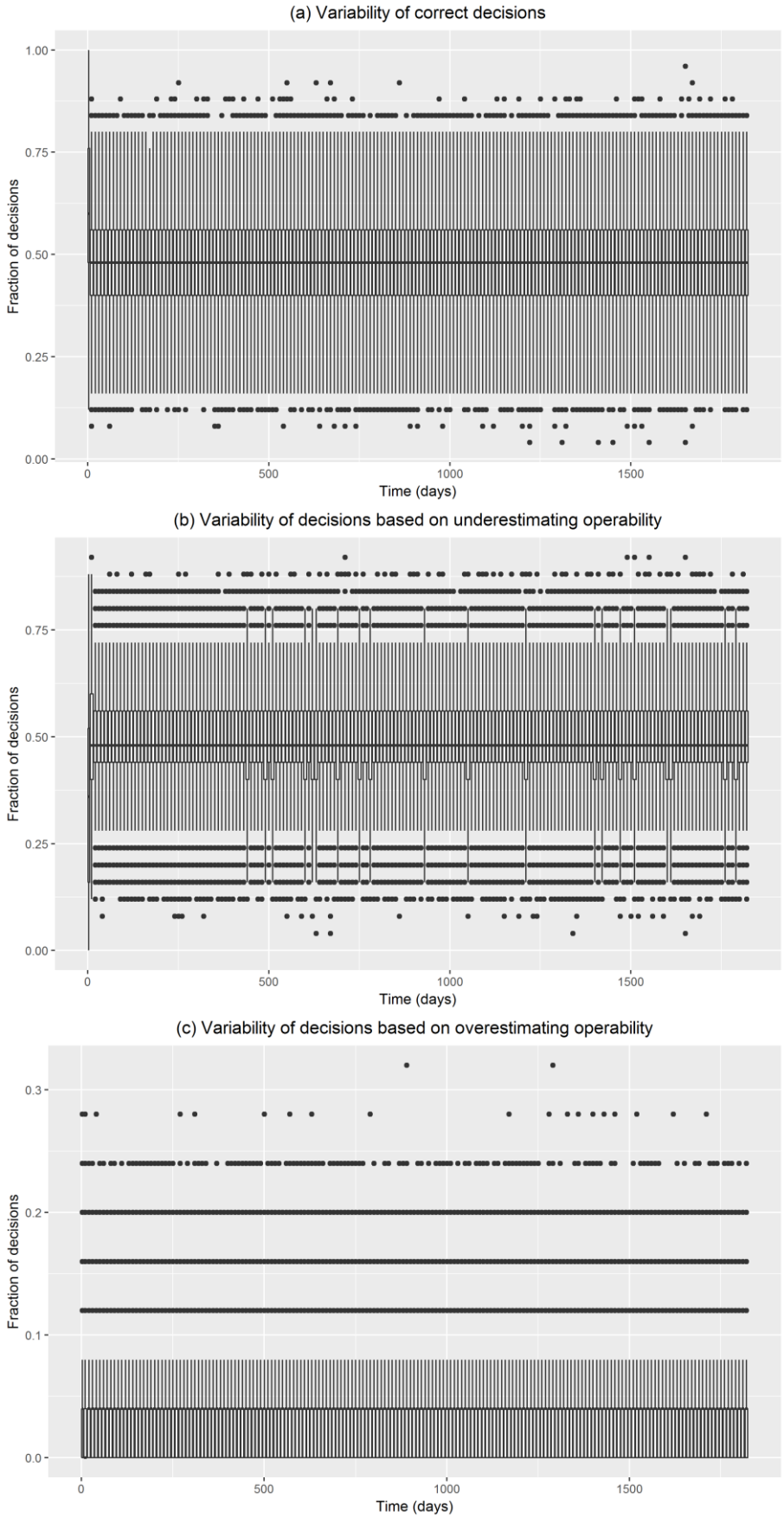


*Figure E-8: Correctness of decisions as a fraction of total decisions made for variability testing*

*Cyberattack effectiveness*

The third set of performance indicators used relate to the effectiveness of cyberattacks. The first performance indicator of this type is the number active attacks. The second performance indicator for the effectiveness of cyberattacks is the average duration of cyberattacks at each point in time. Effectively thwarting unprevented intrusions implies mitigating attacks early. If these numbers differ significantly from the functions formalised for attack detection, these parameters would show deviation beyond expected boundaries. Again, these parameters are likely subject to significant chaos between runs, as similar events might happen at different points in time across model runs. The variability of these parameters can be assessed by verifying whether upper and lower boundaries could be explained by the formalised model and whether the behaviour across 1000 repetitions evens out to a symmetric overview as seen for other performance indicators.

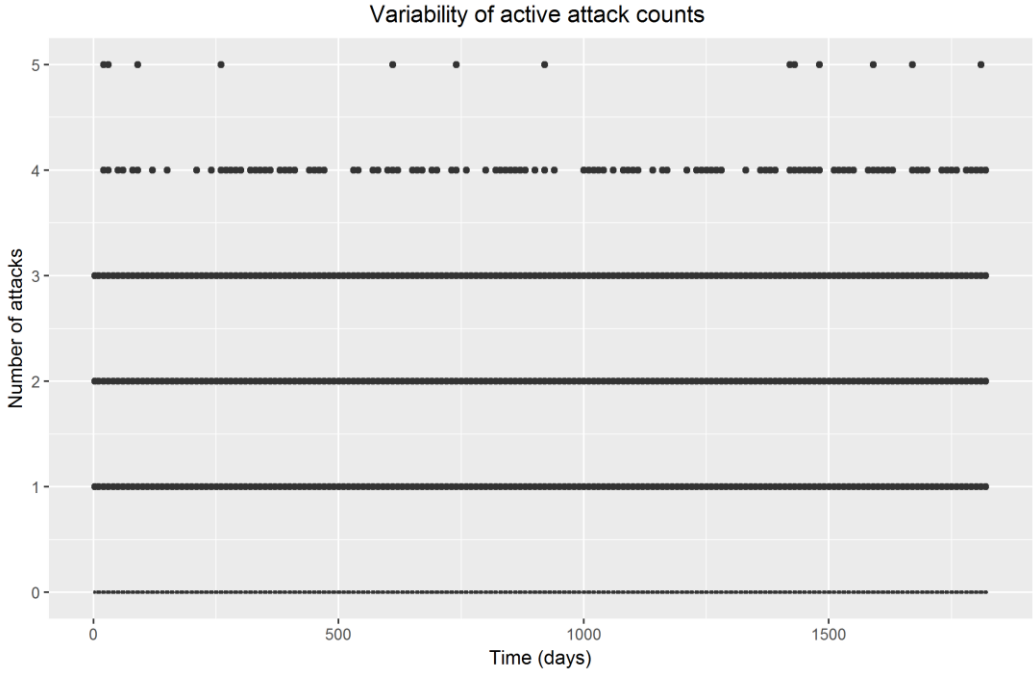


*Figure E-9: Average number of attacks at each time step for variability testing*

The average number of attacks is plotted in Figure E-9 and the average duration is shown in Figure E-10. Both graphs do not show significant variance from 0, as the vast majority of observations are characterised by no active attacks. The full range covered by each boxplot represents 0 attacks, meaning all other observations are outliers. Out of all 1825000 data points, only around 18.7% of observations have at least 1 active attack. While around 1 attack per tick would be expected based on the parameterisation, this number is the result of attempted attacks being thwarted by intrusion prevention systems and therefore not being realised in the model at the end of each tick. Since the number of active attacks is low, the average duration also inherently is tied to 0 for the majority of observations. The subset of attacks with a duration of greater than 0 is even smaller, as there can be active attacks that have just been initialised. This is indeed the case, as there are such attacks in around 14.8% of observations. While this number might suggest some overlap with the same percentage previously identified for inoperable nodes, the underlying number differs (270404 attacks and 271125 inoperable nodes). The variability among these parameters can therefore be logically explained due to successful attack mechanisms and the sheer rarity of loss events due to cyberattacks.

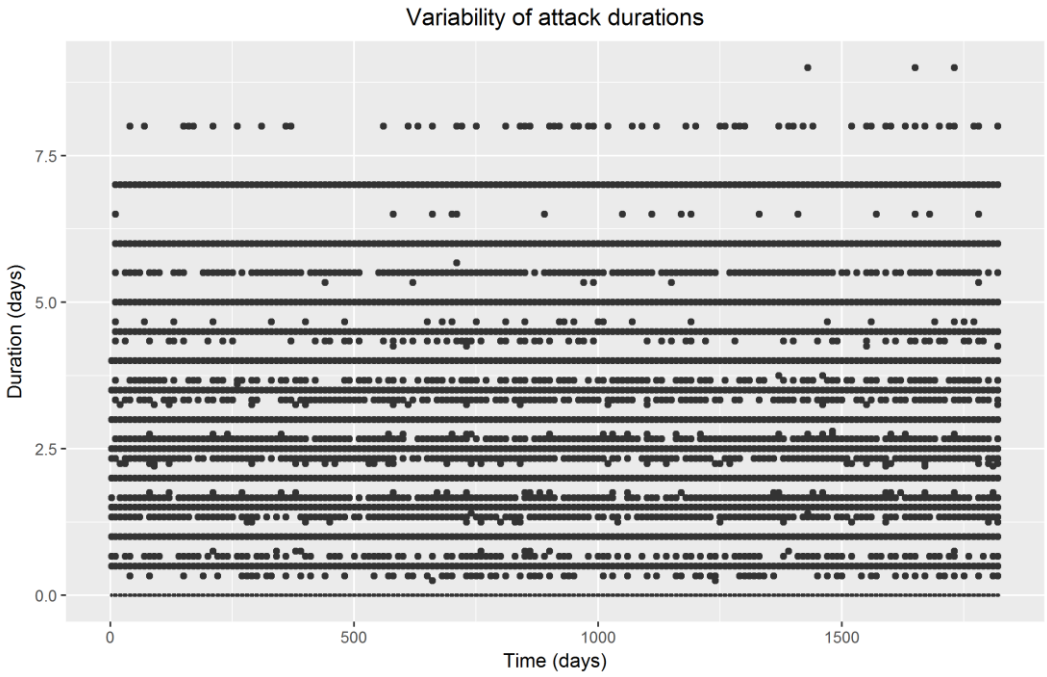


*Figure E-10: Average attack duration at each time step for variability testing*

*Variability between individual repetitions*

In order to establish whether there are substantial differences between individual runs for the same parameter settings, two main paths could be taken: a tabular approach based on descriptive statistics or a visual approach based on additional plots. Since these plots would be rather convoluted, assessing differences across 1000 repetitions, a tabular approach was taken, denoting the mean and standard deviation for each performance indicator across the set of experiments. The set of 1000 repetitions was used to generate a new dataset with 1000 data points, each denoting the average for each performance indicator across an individual run. This dataset was then used to determine the mean and standard deviation for each performance indicator across the overall set of repetitions, following the same process that will be used for data analysis in chapters 7 and 8.

The results of this analysis are shown in Table E-1. The table shows that no performance indicator results in unreasonable standard deviations, as the value range across the set of 1000 repetitions seems rather stable. Only one performance indicator has a seemingly noteworthy standard deviation compared to the mean, being the cumulative extent of losses over the course of a model run. However, these were also tracked through contemporary losses experienced for each time step, which does not result in similar variations due to the exponential nature of standard deviations. While seemingly significant, the standard deviation only makes up for 6.1% of the mean, and does therefore not change the interpretation of model variability. Model variability, as initially expected, shows variation in terms of model-specific events happening at different time steps, for example due to an attack being created or a raised false alarm for defenders. However, since chaos only affects whether such events happen at specific time intervals, their impact on model runs is minor. Variations in model behaviour across a set of model repetitions with equal parameter settings are not likely to affect interpretation of data analysis outcomes.

*Table E-1: Means and standard deviations for all performance indicators across 1000 repetitions for variability testing*

| *Performance indicator* | *Mean* | *Standard deviation* |
|---|---|---|
| *Cumulative losses* | 5566.92 | 343.53 |
| *Current losses* | 0.12 | 0.0075 |
| *Number of normal nodes* | 10.79 | 0.42 |
| *Number of stressed nodes* | 14.03 | 0.40 |
| *Number of inoperable nodes* | 0.18 | 0.054 |
| *Impact assessment deviation* | -0.059 | 0.0011 |
| *Fraction of 'nothing' decisions* | 0.39 | 0.0073 |
| *Fraction of alleviation decisions* | 0.60 | 0.0064 |
| *Fraction of retention decisions* | 0.0072 | 0.0021 |
| *Fraction of correct decisions* | 0.48 | 0.013 |
| *Fraction of overestimated decisions* | 0.030 | 0.0015 |
| *Fraction of underestimated decisions* | 0.49 | 0.014 |
| *Number of active attacks* | 0.21 | 0.027 |
| *Attack duration* | 2.66 | 0.098 |

## Timeline sanity testing

The second form of multi-agent testing involves assessing several parameters for a small number of model repetitions, in this case three repetitions drawn from the dataset used for variability testing. Since this step serves mainly as a tentative check on whether model behaviour is valid, only a select few parameters will be assessed. This step was initially mainly conducted as part of tracking agent behaviour and was consistently applied throughout model implementation.

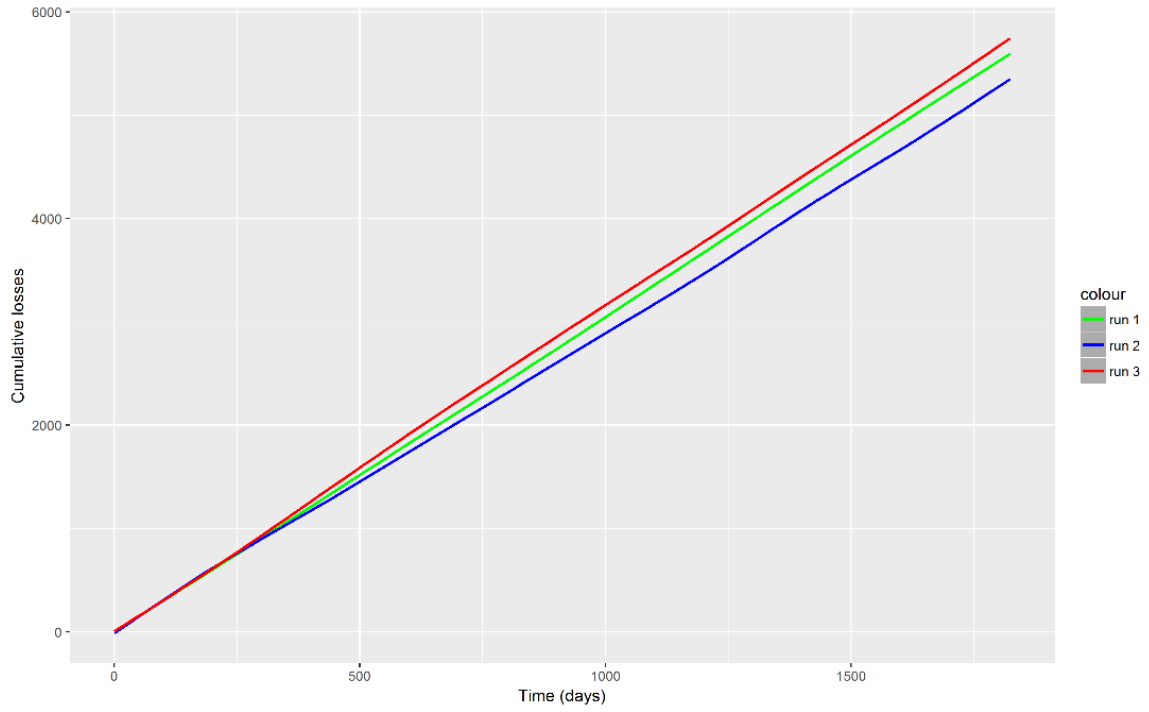


*Figure E-11: Cumulative losses over time for timeline sanity testing*

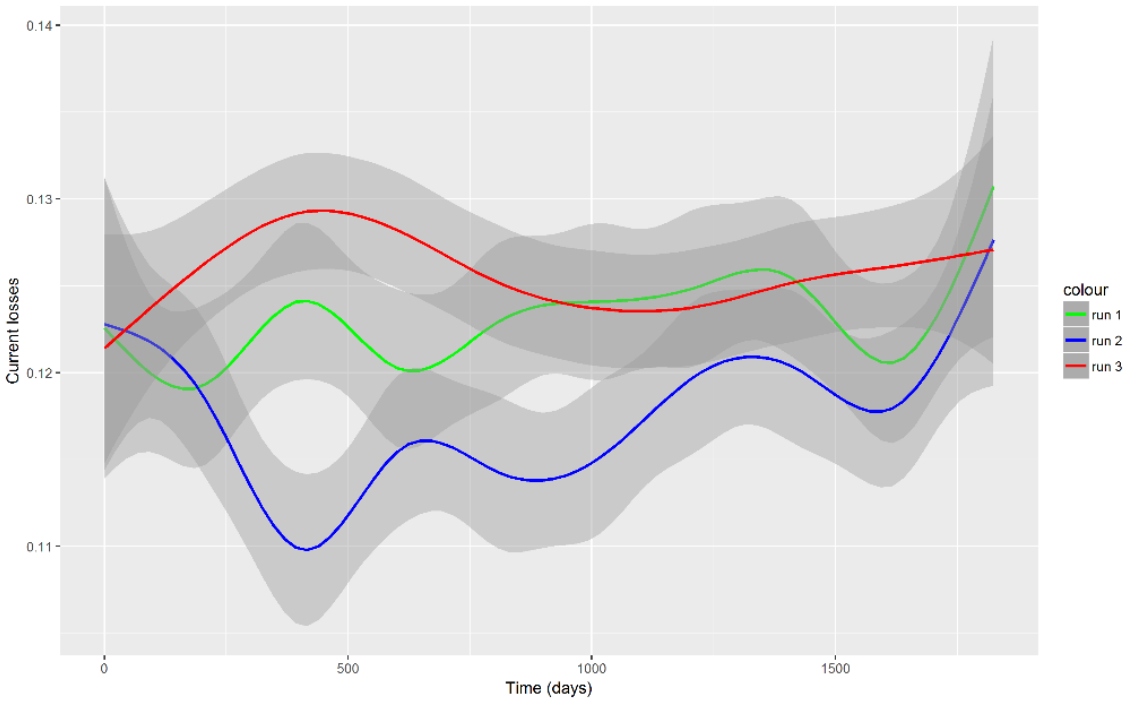


*Figure E-12: Current losses at each time step for timeline sanity testing*

As can be seen in Figure E-11 and Figure E-12, the loss function follows a continuous near-linear growth pattern. Deviations in current loss per time step are rather minor, and when smoothed out, as done for the plot in Figure E-12, show only slight persistent tendencies. The grey areas around each mean graph indicate the 95% confidence interval for the graph. In order to explain these tendencies, Figure E-13, Figure E-14 and Figure E-15 depict the number of nodes for each status at each time step. The changeable nature of each model run can be derived from these plots, highlighting how the symmetrical observations from variability testing emerges. Changes within individual model runs occur at chaotic time steps, but do not significantly contribute to differences cumulatively observed over the course of 1000 repetitions. Upward trends in current losses, for example early on in run 3 in Figure E-12 can be explained by increases in the number of inoperable and stressed nodes as compared to normal nodes shown in Figure E-15. Similarly, the downward trend for losses in run 2 early on in Figure E-12 corresponds with virtually no inoperable nodes and a relative growth in the number of normal nodes. These behavioural tendencies operate as expected and as desired, and to this extent the timeline sanity tests for the model can be confirmed.


*Figure E-13: Node status per time step for run 1*

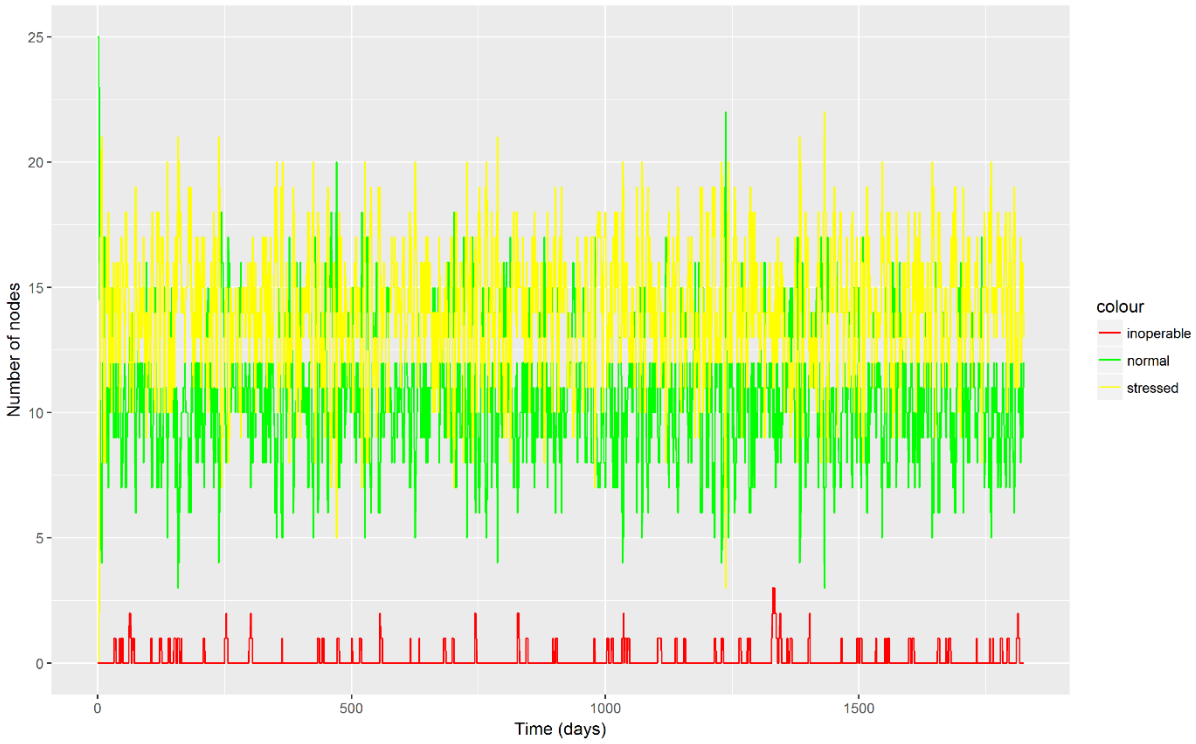


*Figure E-14: Node status per time step for run 2*

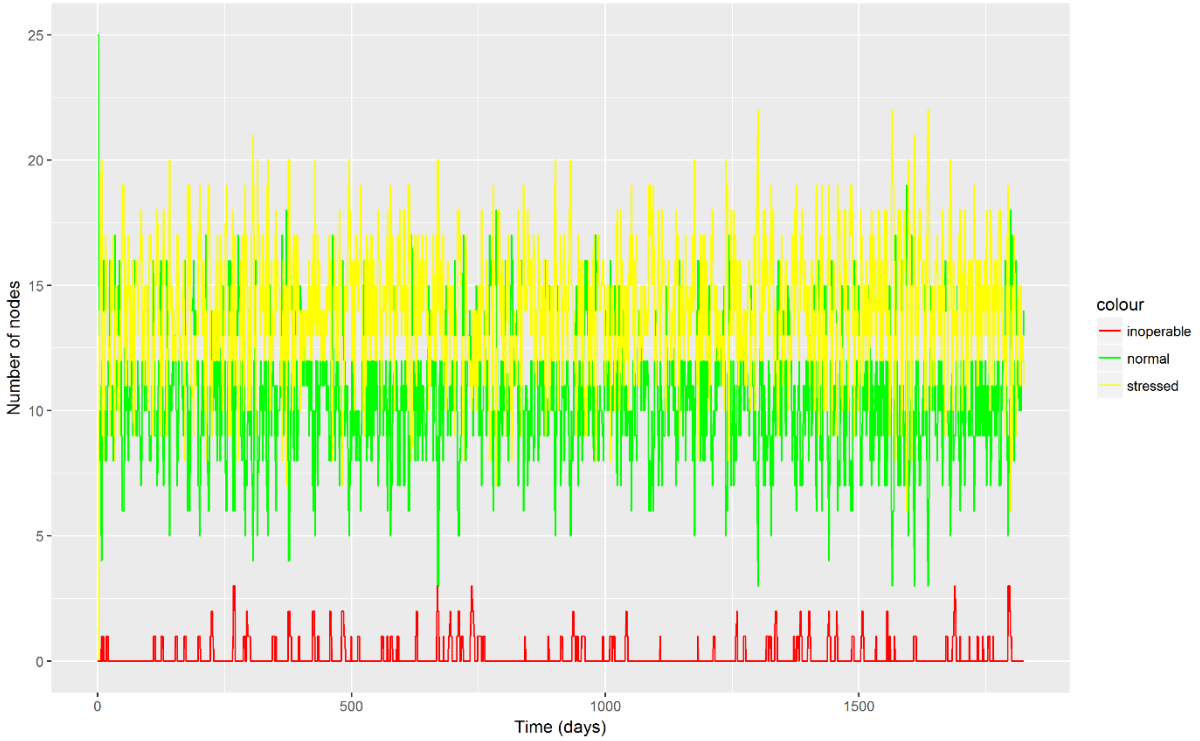


*Figure E-15: Node status per time step for run 3*

# Appendix F: Model parameterisation

This appendix denotes and details parameter values selected for the model. Following the *evaludation* approach by Augusiak et al. (2014), each parameter is assigned a quality indicator on a low/medium/high scale. Several variables cannot be assigned a quality, as these are parameters used for model visualisation (node-radius) or are values the model is based around, such as *Number-of-nodes*. A distinction was made between model parameters, attacker parameters and experimental design parameters. Model parameters impact values used by several agents in their interaction, for example the power assigned with a certain attack. Attacker parameters incorporate the predefined attacks and determine which attacks can be conducted and are fixed by design. Experimental design parameters are used for experimentation and are discussed in more detail in chapter 7.

*Table F-1: Model parameters, values, quality of data and justification*

| *Parameter* | *Value* | *Quality of data* | *Justification* |
|---|---|---|---|
| *Attack-frequency* | 0.050 | High | On average, 20% of critical infrastructure systems are attacked each month (Kaspersky Lab ICS CERT, 2017). That means at a minimum there would be 5 attacks on unique targets per month. Adding slight leeway would result in around one attack per tick. Given the selected number of 20 attackers, each attacker will attack once on average every 20 days. Combined with the number of nodes in the model, this results in roughly 1 attack per tick. |
| *Number-of-attackers* | 20 | N/A | Combining the frequency of attacks to an ecosystem, selecting 20 attackers still yields dynamic interaction and distributed attack origins while not increasing computational requirements significantly. |
| *Number-of-nodes* | 25 | N/A | The ecosystem is built around a system that contains a substantial number of nodes, connections and dependencies. On an ecosystem-level, 25 nodes allow for sufficient variety of interaction while maintaining representativeness for typical critical infrastructure networks shown in Pederson et al. (2006). |
| *Worm-spread-likelihood* | 0.25 | Low | While there are ample examples of cyberattacks on critical infrastructures infecting multiple systems, it is essentially impossible to define a universal probability for worms to spread to other targets at each time step. This value was assumed after iterative testing to assess when model performance looked meaningful. This parameter should be varied heavily during experimentation. |
| *Attack1-power* | 0.35 | Medium | Represents the *disruptive malware* attack discussed in chapter 3. The value of the attack is derived from the relative impact on infrastructure operation from attacks listed in Miller and Rowe (2012). |
| *Attack1-worm* | True | N/A | |
| *Attack2-power* | 0.55 | Medium | Represents the i*nfrastructure blackout* attack discussed in chapter 3. The value of the attack is derived from the |

| | | | |
|---|---|---|---|
| *Attack2-worm* | False | N/A | relative impact on infrastructure operation from attacks listed in Miller and Rowe (2012). |
| *Attack3-power* | 0.6 | Medium | Represents the *infrastructure asset destruction* attack discussed in chapter 3. The value of the attack is derived from the relative impact on infrastructure operation from attacks listed in Miller and Rowe (2012). |
| *Attack3-worm* | True | N/A | |
| *Criminal-distribution* | 1 | N/A | The relative distribution of each type of attacker among the total number-of-attackers. By default these are assumed to be equally present, all values are set to 1. |
| *Terrorist-distribution* | | | |
| *Adversary-distribution* | | | |
| *Node-radius* | 4 | N/A | Visual modifier that impacts how far apart nodes are created. Has no real-world counterpart of relevancy to experiments. |
| *Physical-impact* | Random between 0 and 1 | Low | Both *physical-impact* and *economic-impact* exist as parameters to incorporate optimisation-based targeting mechanisms for attackers. The extent of losses itself does not matter – only relative performance between defensive strategies yields interesting results. |
| *Economic-impact* | Random between 0 and 1 | Low | See *physical-impact*. |
| *Number of connections* | 1.5/node | Medium | Based on the connectivity model by Pederson et al. (2006). Implemented as a conceptual element to facilitate worm infection capabilities. |
| *Number of dependencies* | 1/node | Medium | Based on the dependency model by Pederson et al. (2006). Implemented to facilitate the possibilities of cascading failures. Can be changed further to incorporate different structures for dependencies or infrastructure networks in general. |
| *Dependency weighting* | 0.3 | Low | Guesstimate based on the level of operability and general effects of cascading failures. Based on the multitude of literature discussed in section 2.1. |
| *User-traffic-frequency* | 0.4 | Low | Guesstimate based on the level of operability and typical effects for failing to classify traffic correctly (Vasilomanolakis et al., 2015). |
| *User traffic criticality* | 0.4 | Low | See *User-traffic-frequency.* |
| *Alleviation-duration* | 7 | Low | Guesstimate based on iterative model performance. Estimated based on the requirements for responding in relation to attack powers. In reality, responses take significantly shorter (Department of Homeland Security, 2015). Changing the time interval to allow for shorter time steps would increase computational requirements exponentially. Since there is no need to produce tangible, real-world representations of attack-defence scenarios, the increased time for this is not problematic, |

| | | | |
|---|---|---|---|
| | | | so long as it remains consistent with other model parameters. |
| *Retention-duration* | 4 | Low | See *Alleviation-duration.* |

*Table F-2: Attacker parameters*

| *Attacker type* | *Physical-preference** | *Economic-preference** | *Attack1?* | *Attack2?* | *Attack3?* | *Knowledge* |
|---|---|---|---|---|---|---|
| *Cybercriminal* | 0.1-0.3 | 0.9-1 | x | x | | Low |
| *Cyberterrorist* | 0.9-1 | 0.7-0.9 | | x | x | Medium |
| *Foreign adversary* | 0.9-1 | 0.4-0.6 | x | x | x | High |

* *Physical-preference* and *economic-preference* are both used to add variety to attack targeting behaviour and do not necessarily have real-world counterparts. These are based on the motivation for attackers discussed in section 3.2 and primarily work to discern between targets.

*Table F-3: Defensive strategy configuration parameters, values, quality and justification*

| *Defensive strategy configurations* | *Value* | *Quality of data* | *Justification* |
|---|---|---|---|
| *Prevention sensitivity* | Anomaly-based: 0.95<br>Signature-based: 0.8 | High | These parameters and their meaning are based on the taxonomy of intrusion detection and prevention systems by Patel et al. (2013). However, the authors do not specify unambiguous values to be used for such mechanisms. Instead, values concluded by Mansour, Chehab, and Faour (2010) are applied. They state that average false negative and false positive rates for both prevention and detection range between 5-15%, depending on the method applied. True negative and true positive rates are derived from these values and deviate between such values. The average success rate for attacks on networked control systems is around 10%, roughly corroborating these distributions (Verizon, 2015). The quality of the data is considered high despite not directly linking the parameters to direct implementations of such features, as experimentation is exploratory in nature. The aim is to find emergent patterns for several defensive strategies as opposed to designing a specific intervention for an existing system. |
| *Prevention specificity* | Anomaly-based: 0.8<br>Signature-based: 0.95 | High | |
| *Detection sensitivity* | Anomaly-based: 0.95<br>Signature-based: 0.8 | High | |
| *Detection specificity* | Anomaly-based: 0.8<br>Signature-based: 0.95 | High | |
| *Alleviation threshold* | 0.7 | Medium | Values derived from the central degree of operability of a node by Setola and Theocharidou (2016), combined with typical attack strengths and likelihood to classify attacks correctly. These values differ per defensive strategy to account for differences in accuracy for situational awareness. |
| *Retention threshold* | 0.3 | Medium | |

# Appendix G: Model exploration

This appendix contains analysis on the outcomes of experimentation. Only the three most influential parameters are described following the process described in section 7.2.

## G.I Dependency weighting

The overall impact of dependency weighting is supposed to primarily affect the total extent of losses inflicted to the ecosystem. Attack-induced inoperability in nodes translates over to dependent nodes, directly causing further inoperability. More heavily weighted dependencies further exacerbate the degree to which losses are incurred. This is corroborated by the density plot shown in Figure G-1, with higher values for dependency weighting directly leading to higher and less densely concentrated losses. In simulations with heavily weighted dependencies, slight variations are exaggerated, causing a wide spread in the behaviour shown. The behaviour shown here shows sensitivity, but not to an unreasonable extent. The implementation of dependency weighting as a direct link between the operability levels is conceptually corroborated, but values used in the model are uncertain by their nature (Setola & Theocharidou, 2016). Effective defensive strategies should account for robustness across the variety of scenario parameter ranges.

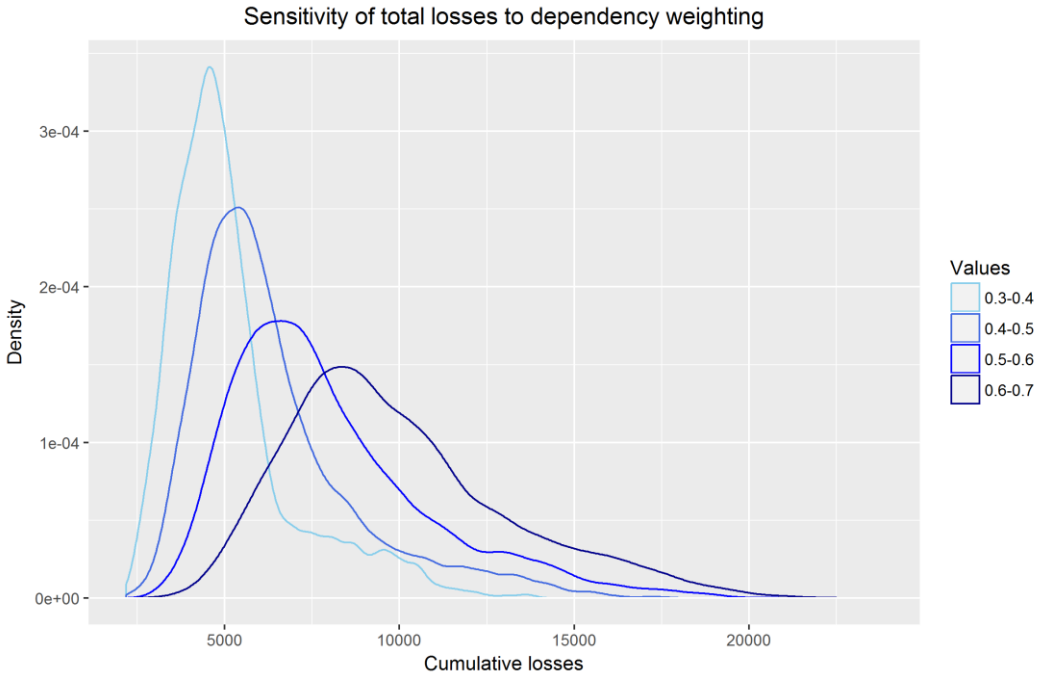


*Figure G-1: Effects of dependency weighting on total losses*

While the effects of different values for dependency weighting on total losses at the end of a simulation are noticeable, the effects on correctness of decision-making are not likely to vary significantly. The reason for this is rather straightforward: dependency weighting only affects the external operability component of a node and does not interfere directly with internal impact assessment. However, decisions are made based on the overall impact perception, which is partially affected by external inoperability. Figure G-2a shows two distinct clusters with identical patterns of observations. Both the higher and lower distributions of observations highlight a slight increase in the correctness of decisions made for heavier dependency weightings, while the relative spread is larger for the cluster of fewer correct decisions. Conversely, Figure G-2c shows fewer overestimations of operability occur for lower dependency weightings. This implies that the unexpected decrease in decision-making accuracy occurs due to a higher frequency of

underestimating the level of operation, corroborated by Figure G-2b. The only logical explanation for this phenomenon is that in some cases, more heavily weighted dependencies lead to correct defensive decisions without correctly assessing internal impact. In those cases, defensive decisions are made based on primarily external perturbation, and by extent are 'accidentally correct'. These variations, while unexpected, are relatively minor and could logically be explained by model formalisms. The existence of two separate clusters of observations is attributable to differences between defensive strategies.

Compared to variance observed for other parameters, the density of impact assessment deviation across several values for dependency weightings, shown in Figure G-3, is relatively stable. Deviations are almost indistinguishable, with one slight exception. As with the distribution of erroneous decisions, lower values show slightly more dense clusters of observations for lower values and less dense clusters for higher values, corroborating what was written above. However, this variance is rather marginal when compared to the effects of dependency weighting on the total extent of losses.

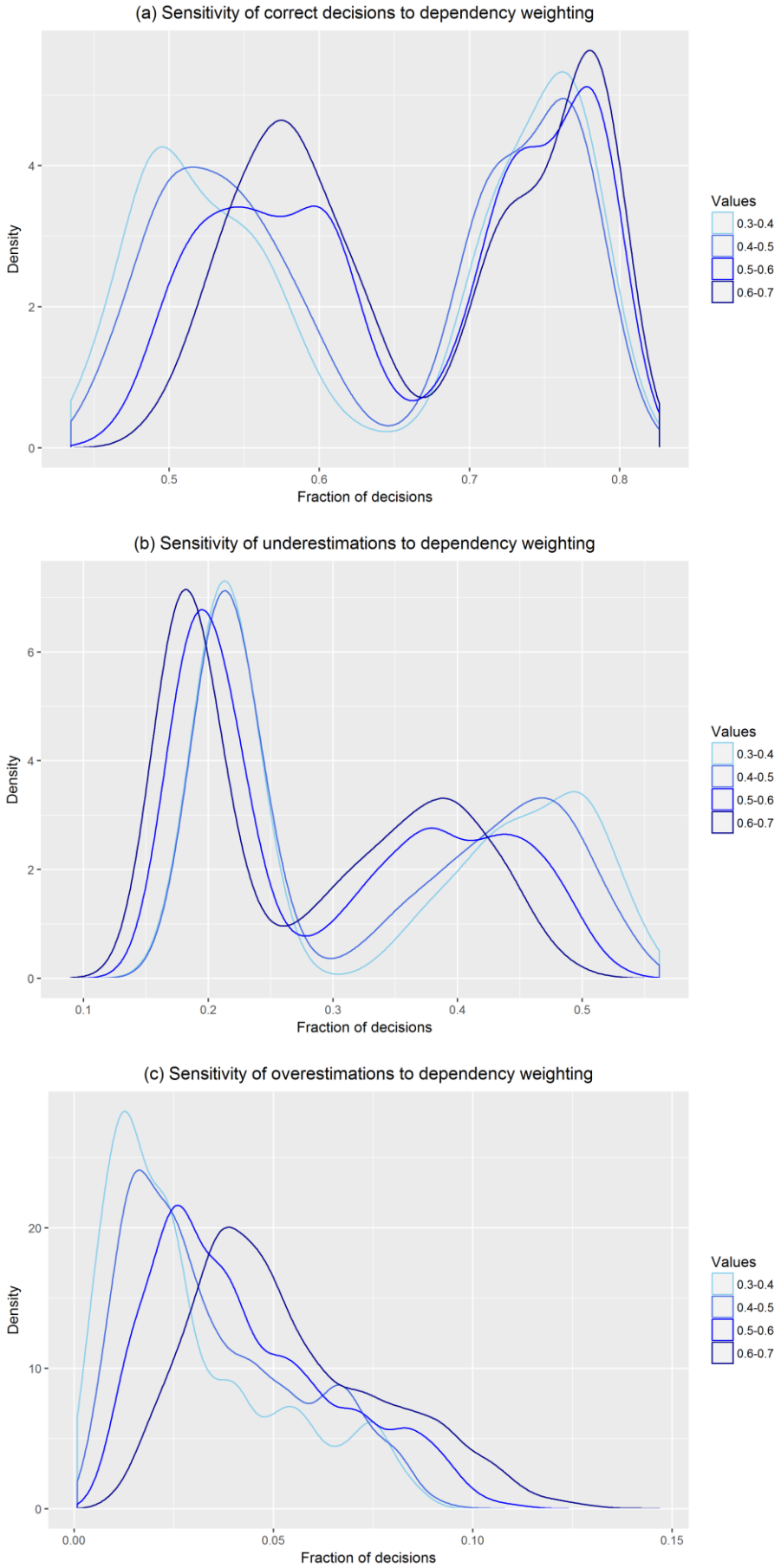


*Figure G-2: Effects of dependency weighting on decision correctness*

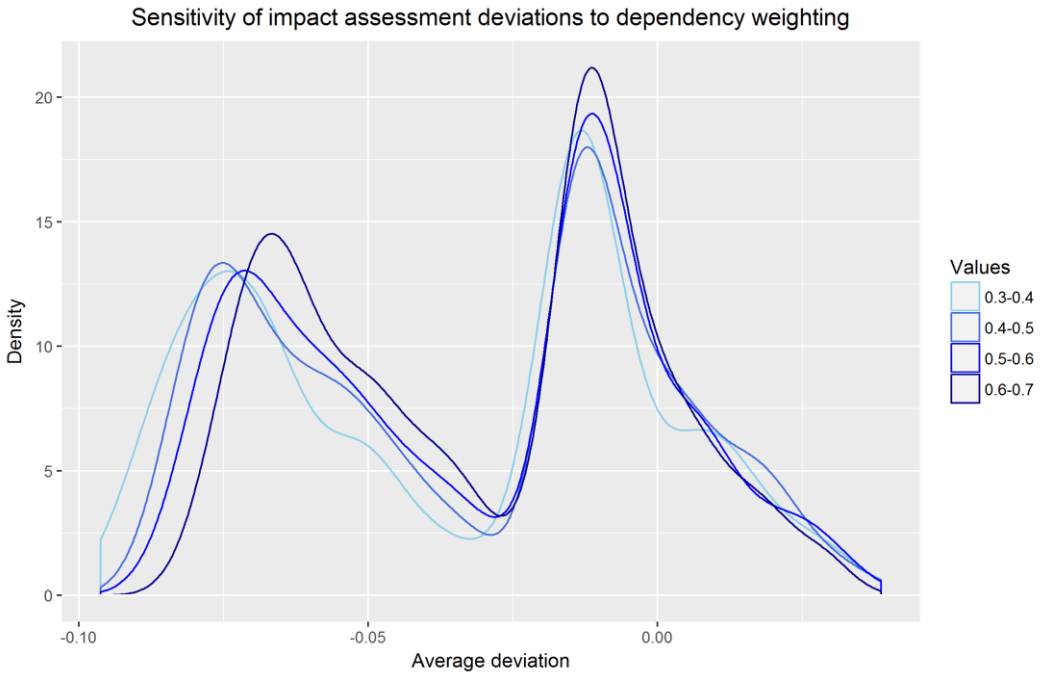


*Figure G-3: Effects of dependency weighting on impact assessment deviation*

## G.II Attack frequency

Variations in values for attack frequency directly affect the presence of threats to the ecosystem, as attacks become a more common occurrence. As shown in Figure G-4, the extent of total losses suffered across repetitions increases as the attack frequency increases. However, when compared to the extent of losses experienced for variations in dependency weighting, these are relatively minor. Interestingly, the highest values for attack frequency show different behaviour, with more widely spread density of losses. This is likely caused by the combination of higher attack frequencies and more powerful attacks or other synergetic parameters, exacerbating the extent of inoperability caused by attackers.

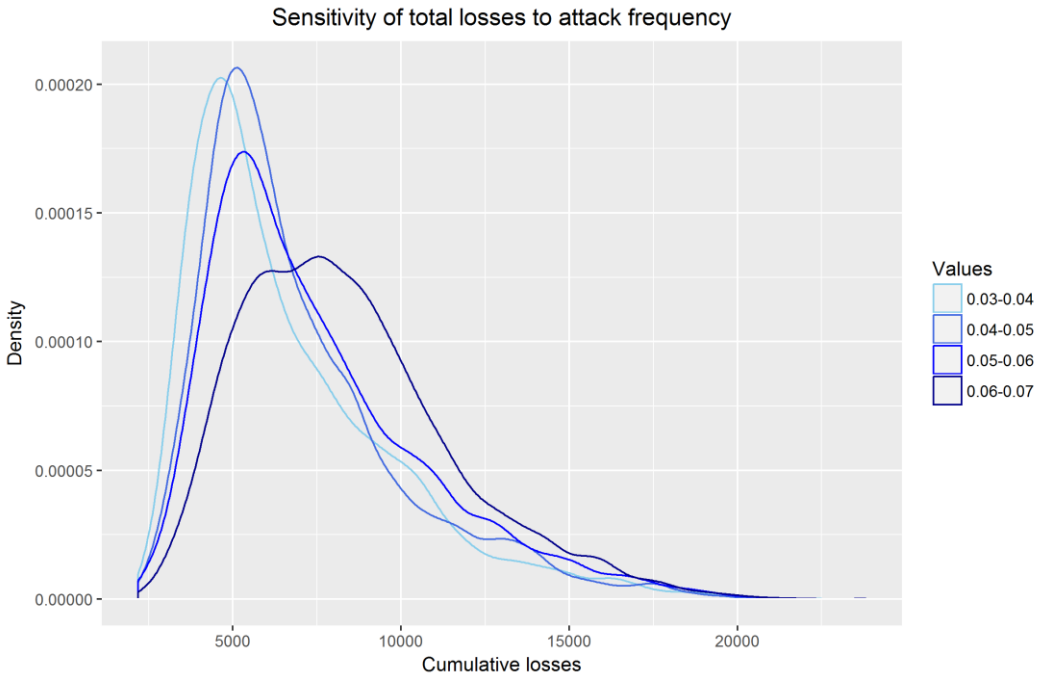


*Figure G-4: Effects of attack frequency on total losses*

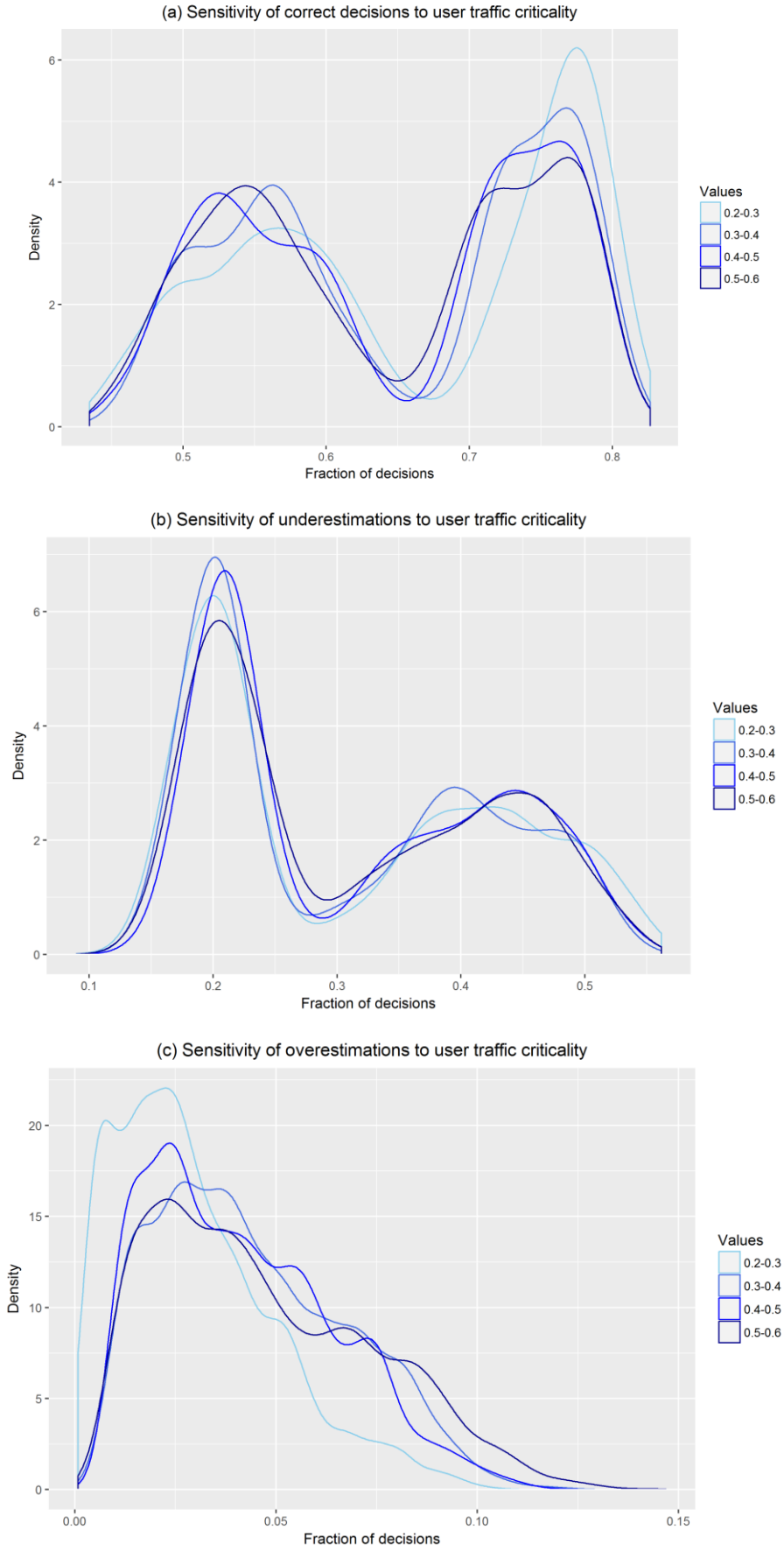
(a) Sensitivity of correct decisions to user traffic criticality
Density
Fraction of decisions
Values
0.2-0.3
0.3-0.4
0.4-0.5
0.5-0.6
(b) Sensitivity of underestimations to user traffic criticality
Density
Fraction of decisions
Values
0.2-0.3
0.3-0.4
0.4-0.5
0.5-0.6
(c) Sensitivity of overestimations to user traffic criticality
Density
Fraction of decisions
Values
0.2-0.3
0.3-0.4
0.4-0.5
0.5-0.6

*Figure G-5: Effects of attack frequency on decision correctness*

Variations in the frequency at which attacks occur only bear moderate consequences for decisions made across the set of experiments. This is in line with expectations, as the metrics shown in Figure G-5 indicate the correctness of decisions as a fraction. As a result, decisions made across a simulation are similar regardless of the frequency of attacks they are exposed to. The one observation is that lower attack frequencies lead to a slightly higher number of underestimations, likely attributable to the reduced presence of 'accidentally correct' decisions, as described previously. The system shows robust behaviour across this parameter set, an indication that overall fluctuations in performance are largely caused by defensive strategies. These observations also hold up for the effects of attack frequency deviations on the density of deviations in impact assessment, shown in Figure G-6.

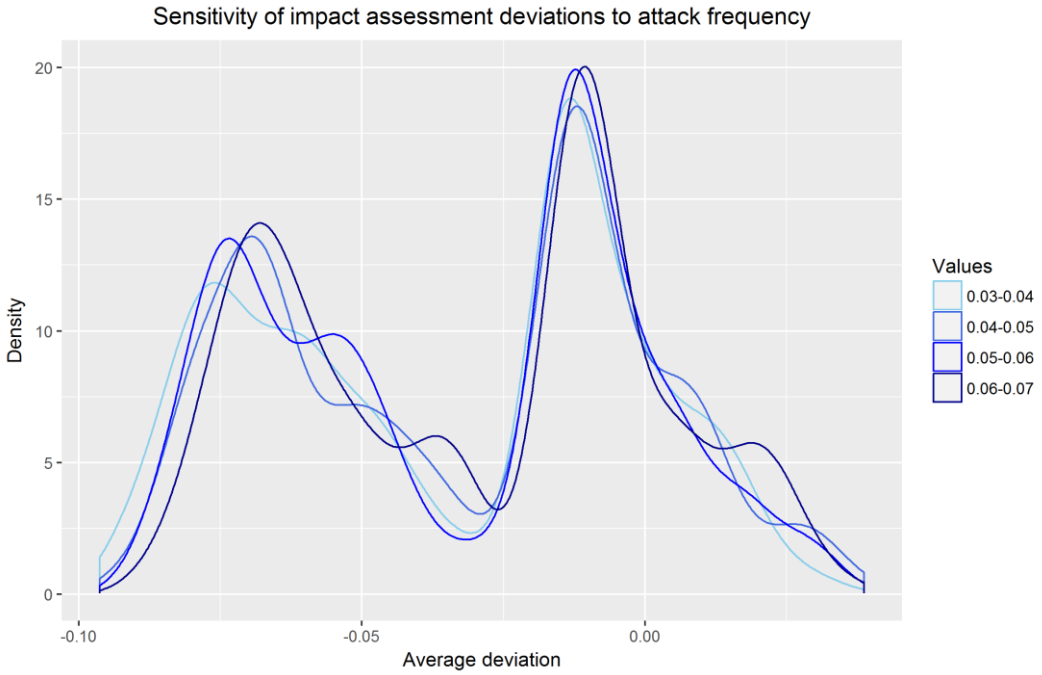


*Figure G-6: Effects of attack frequency on impact assessment deviation*

## G.III Attack powers

The impact of attack powers, as with the previous two scenario parameters, directly affects the interaction that takes place between attackers and defenders. However, the values associated with attacks are also used by defenders in their impact assessment process. If a false positive is generated during intrusion detection, defenders decrease their perceived internal operability by the power associated with the type of attack classified. Higher values for attack powers therefore affect nodes both directly and indirectly. The overall deviations in values for attack powers are relatively low, as their impact is based on historical attacks that disrupted parts of critical infrastructure systems (Miller & Rowe, 2012). Deviations across the dimensions for this parameter are therefore expected to be stable and condensed in comparison to other parameter. The behaviour shown in Figure G-7 displays moderate variation, but mainly trending within similar boundaries. The density plots for the lowest and highest set of attack powers show more variability, as these lines are based on a smaller subset of observations. Since attack powers are a summation of individual attack powers, the odds of all attack powers being selected either below or above average are smaller than less extreme deviations. The resulting density plots are more likely to show sensitive variations due to the smaller

sample size. With this in mind, the density plots for total losses are for the most part considered robust.

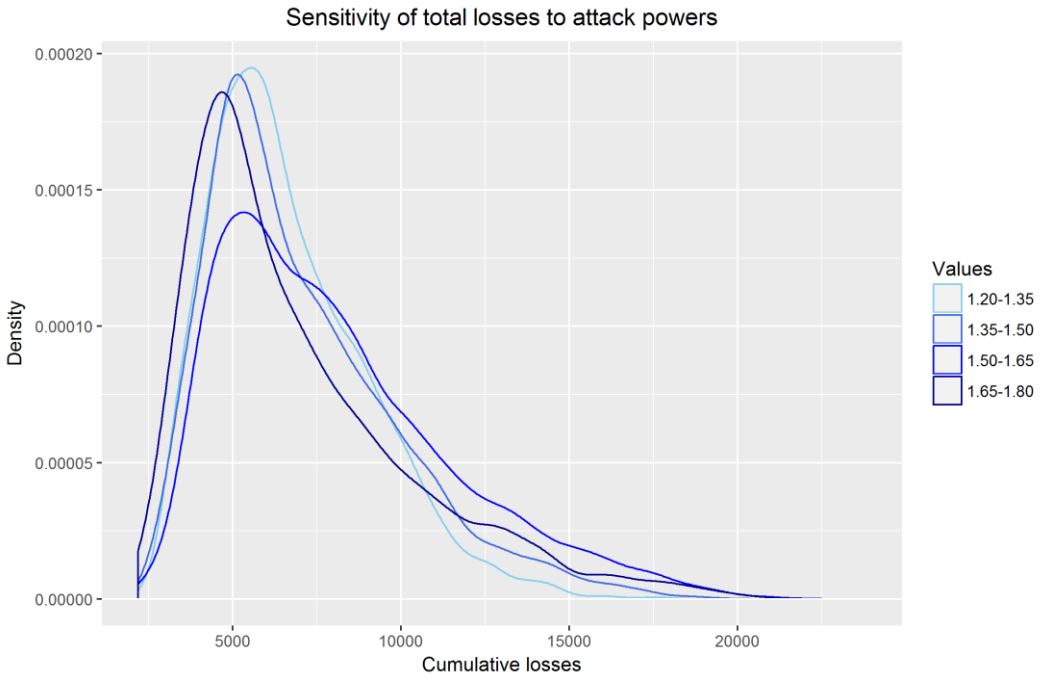


*Figure G-7: Effects of attack powers on total losses*

Besides the total losses incurred during simulations, variations in attack powers could lead to differences in outcomes from decision-making processes. The density plots for correct decisions, decisions based on underestimation of operation and decisions based on overestimation of operation are shown in Figure G-8. Figure G-8a depicts the behaviour of correct decisions made across simulations, showing that the lower two value ranges lead to a noticeable cluster of incorrect decisions on the lower end of the plot. The same pattern appears mirrored in Figure G-8b, indicating that the cluster is caused by underestimating the impact experienced by cyberattacks. Given a lower set of attack powers, the pressure exerted by user traffic becomes relatively more important. Where previous defensive decisions are based around an equilibrium from false negative and false positive detections, this is slightly distorted for lower attack powers. Overall, these changes are marginal and show a desired degree of sensitivity.

Similar behaviour is identified in Figure G-9, which displays the density of impact assessment deviation for each set of attack powers. As with the correctness of decisions, some variation is observed, specifically for the very lowest set of attack powers, which trends higher on average than other value ranges. This corresponds with what was written for the previous set of metrics, as lower attack powers cause a shift in behaviour due to both not detecting impact from erroneously blocked user traffic and this deviation not being sufficiently compensated by false positive detections.

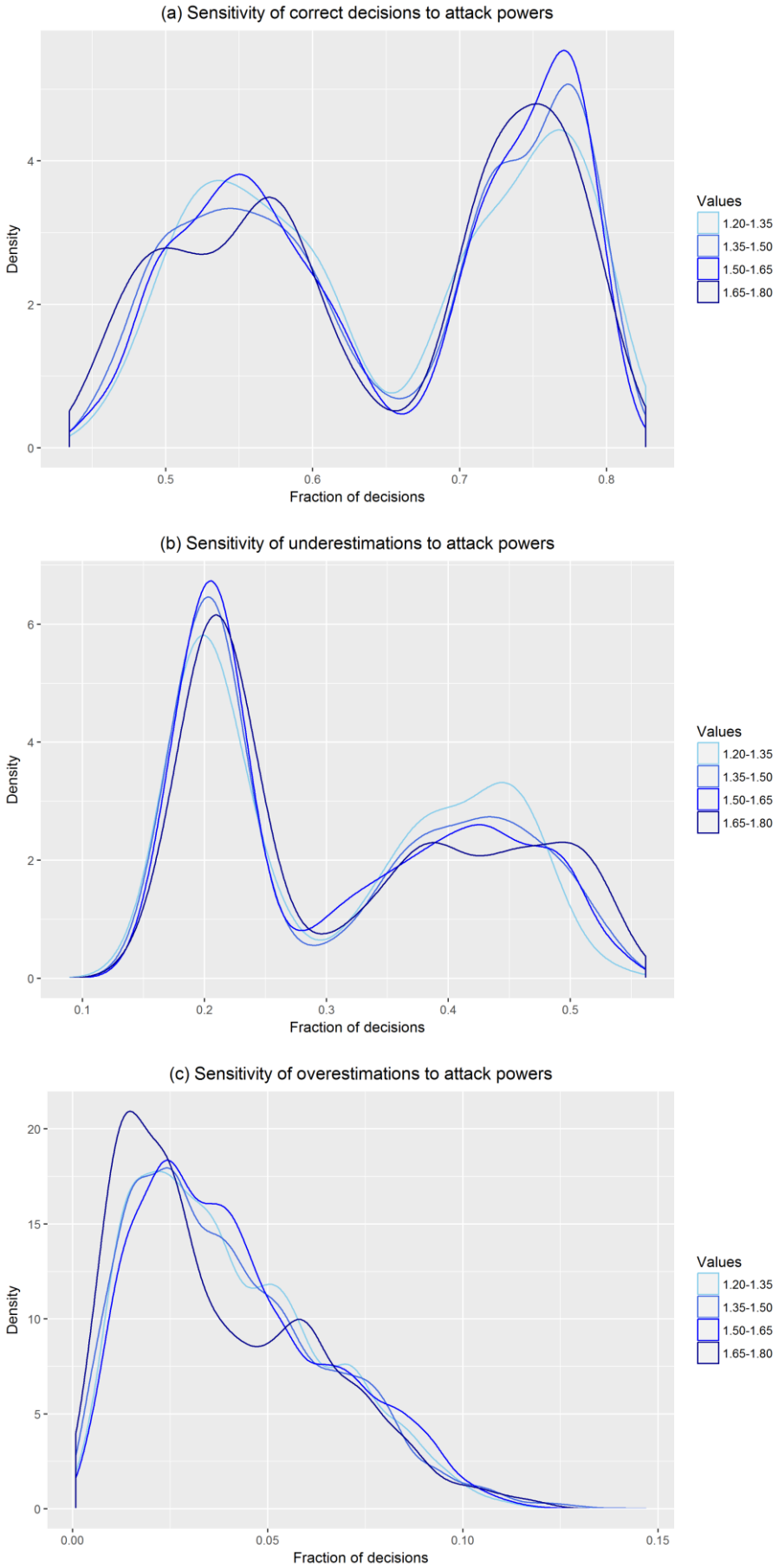


*Figure G-8: Effects of attack powers on decision correctness*

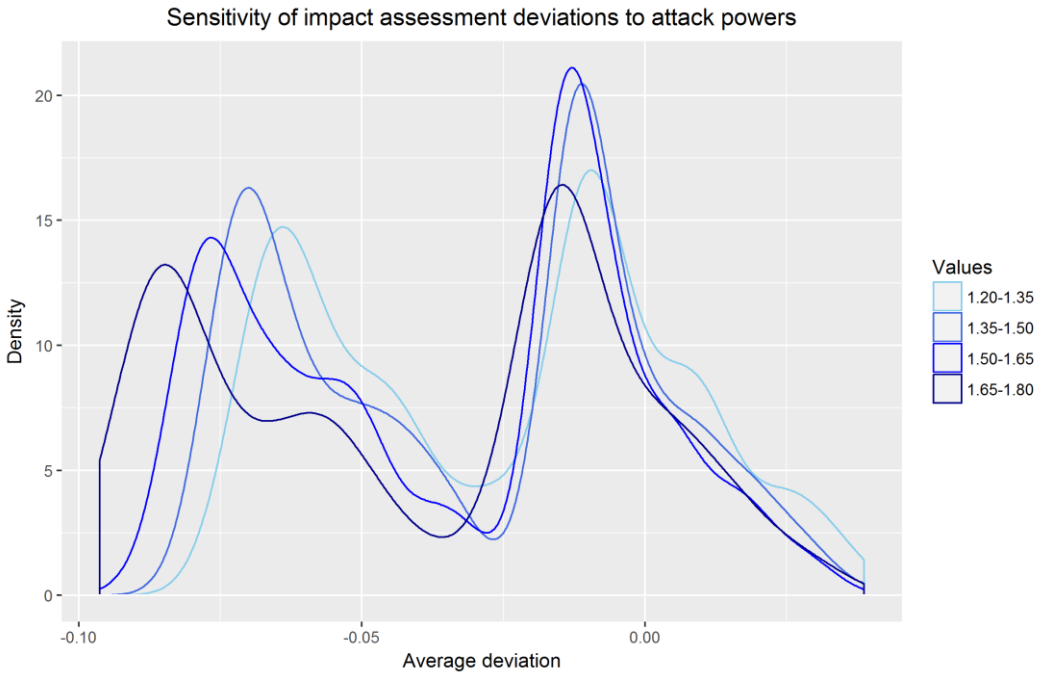


*Figure G-9: Effects of attack powers on impact assessment deviation*

Patterns for other scenario parameters provided little additional insight and are therefore not discussed further.

## G.IV Worm spread likelihood

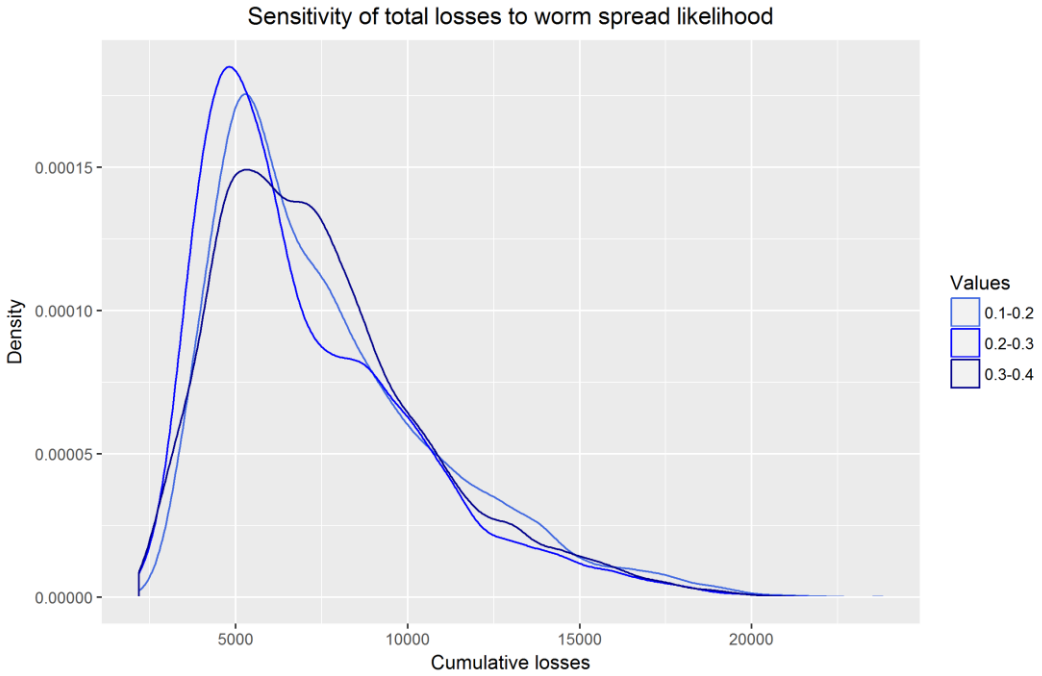


*Figure G-10: Effects of worm spread likelihood on total losses*

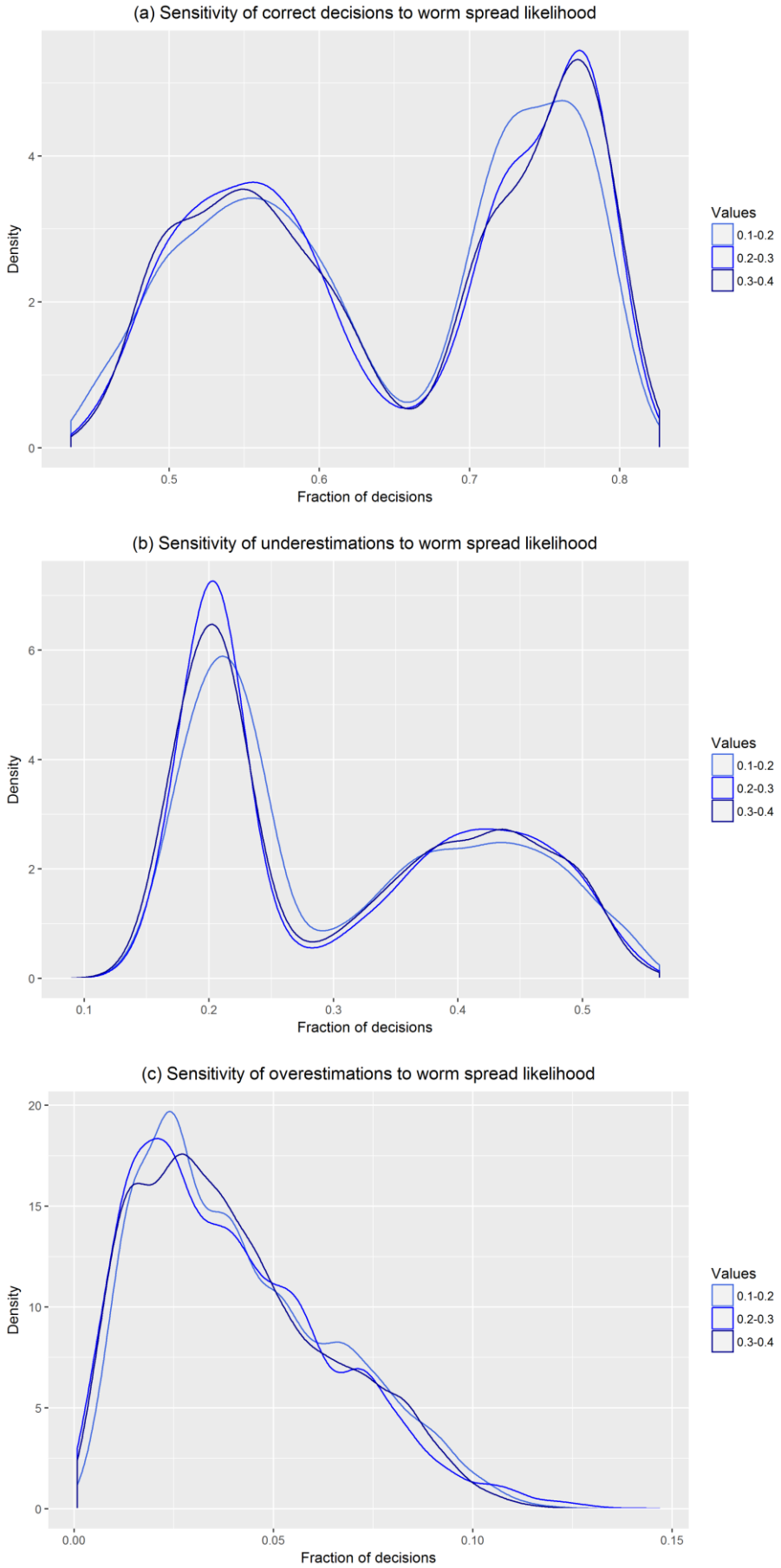


*Figure G-11: Effects of worm spread likelihood on decision correctness*

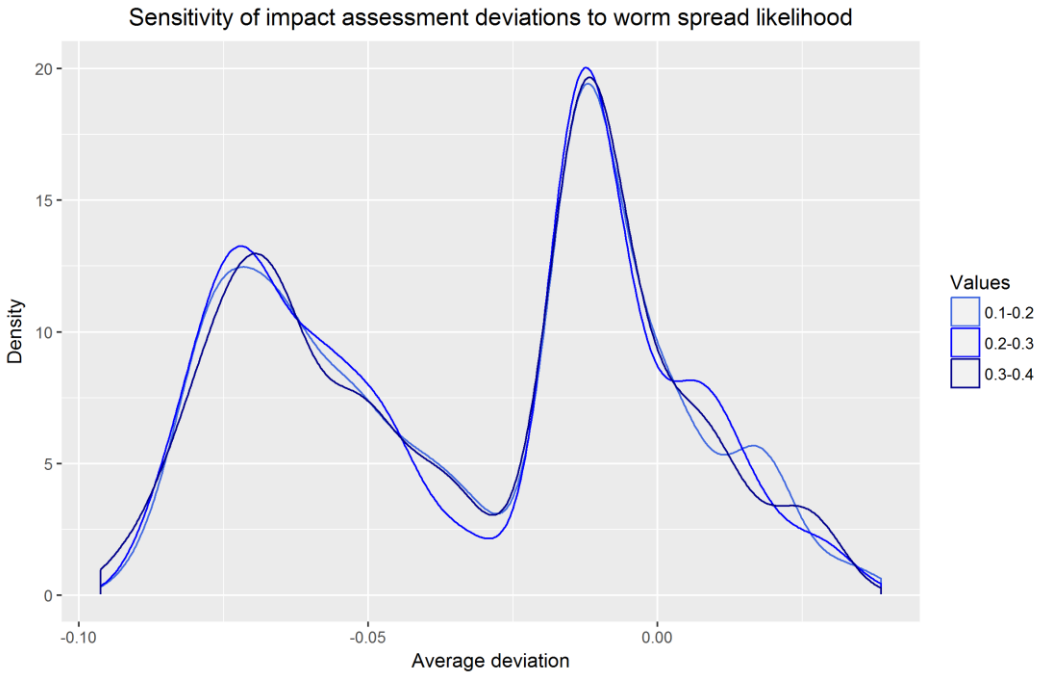


*Figure G-12: Effects of worm spread likelihood on impact assessment deviation*

## G.V Alleviation duration

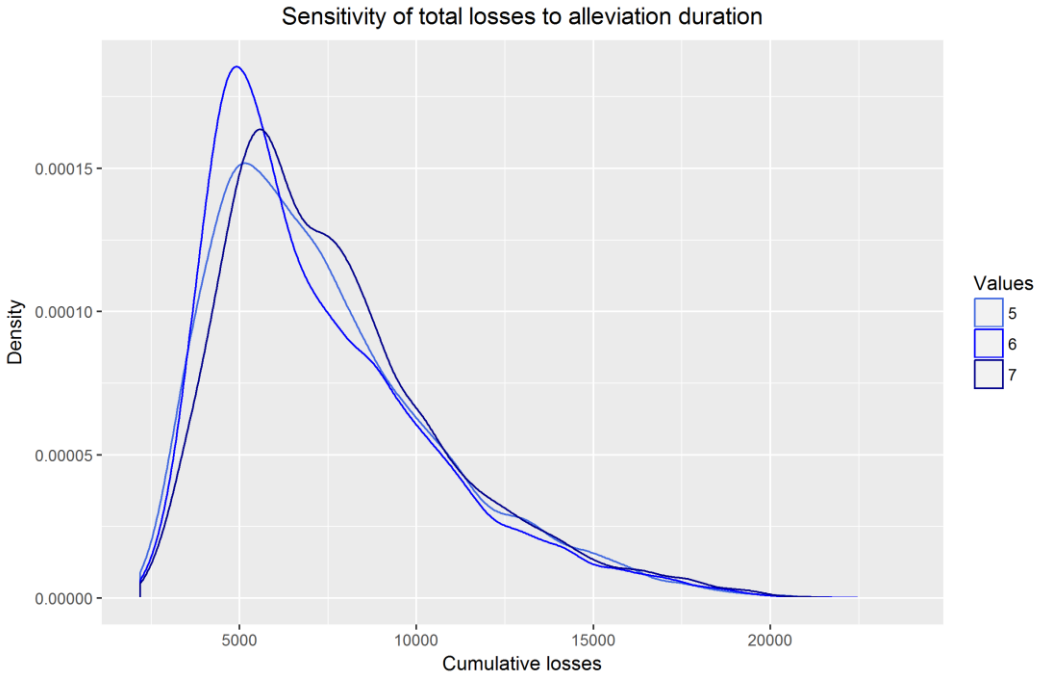


*Figure G-13: Effects of alleviation duration on total losses*

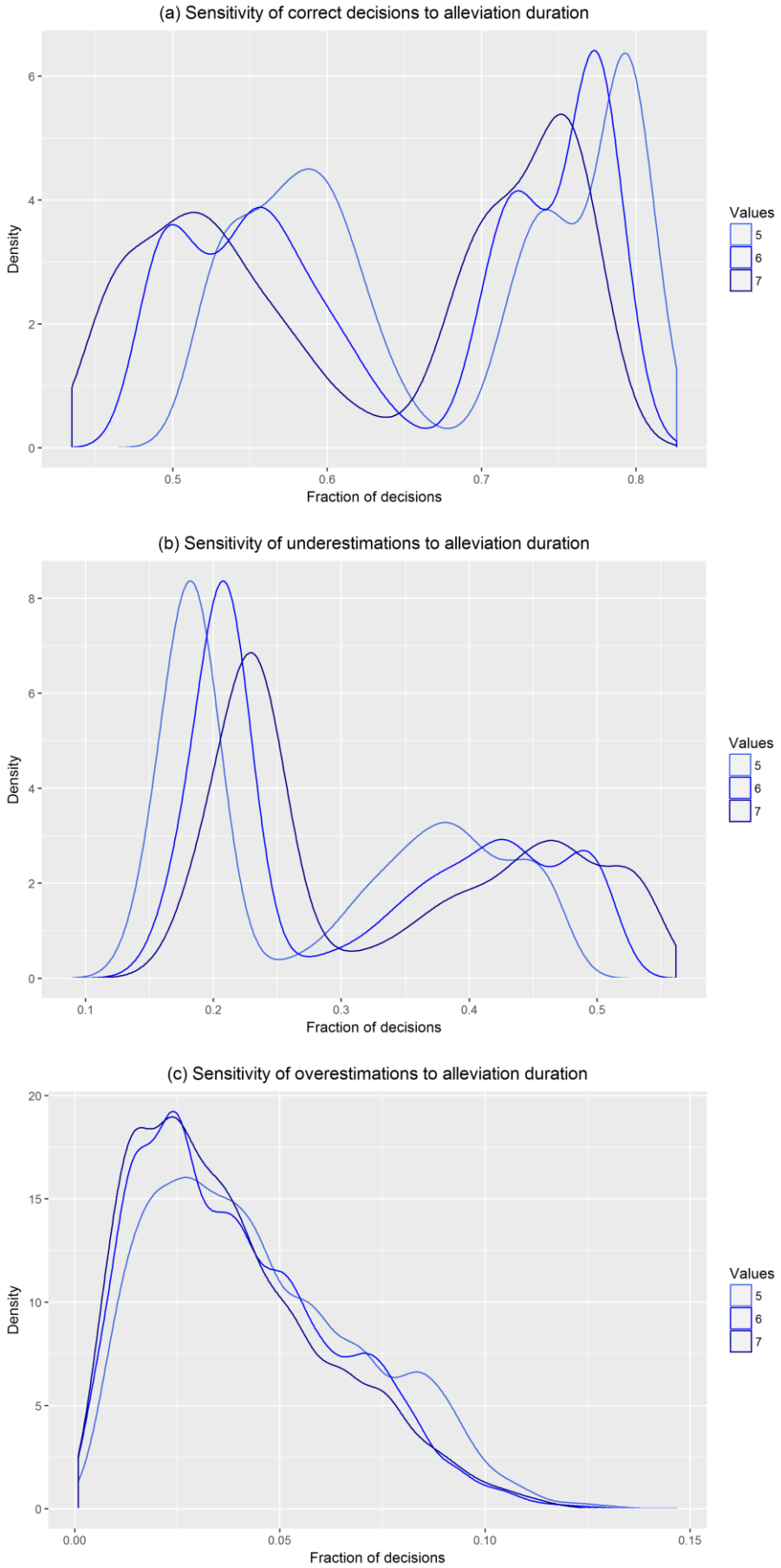


*Figure G-14: Effects of alleviation duration on decision correctness*

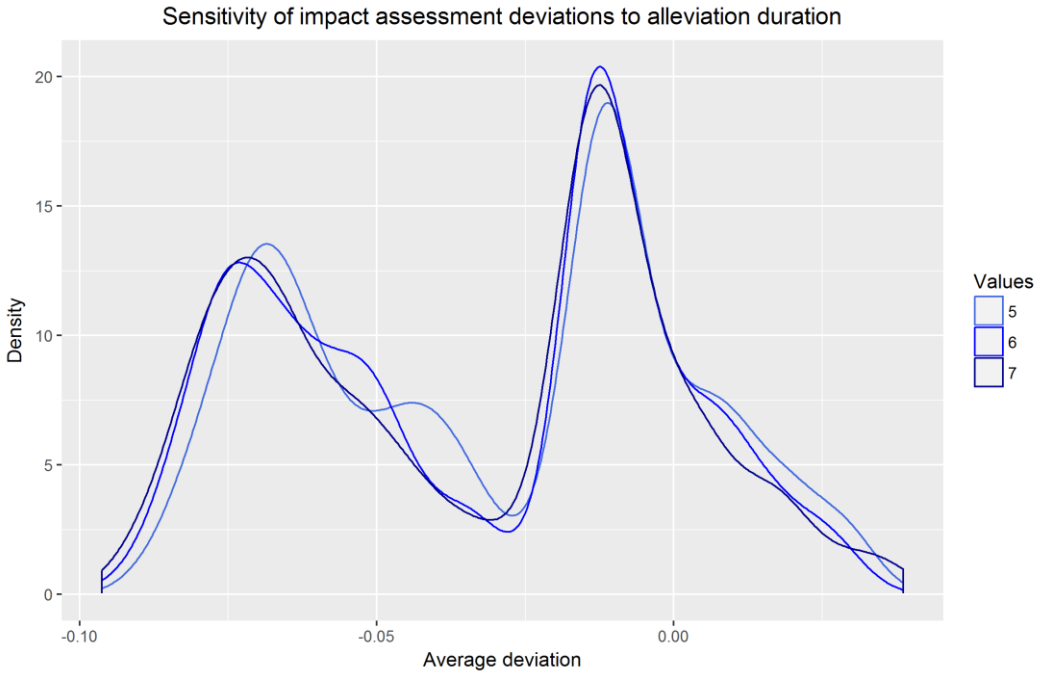


*Figure G-15: Effects of alleviation duration on impact assessment deviation*

## G.VI Retention duration

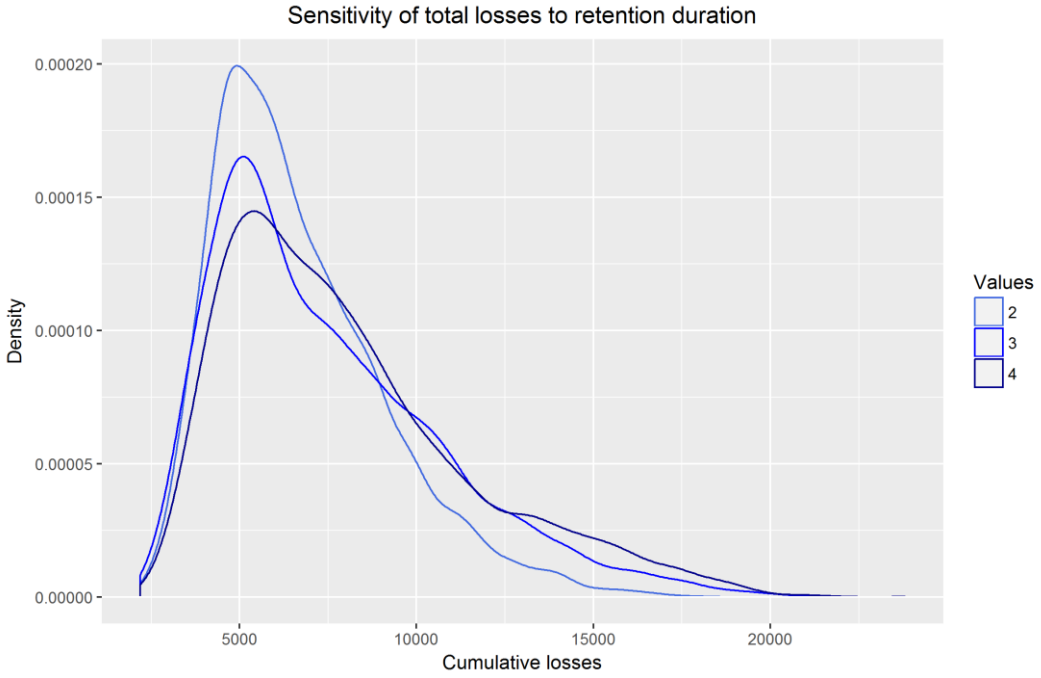


*Figure G-16: Effects of retention duration on total losses*

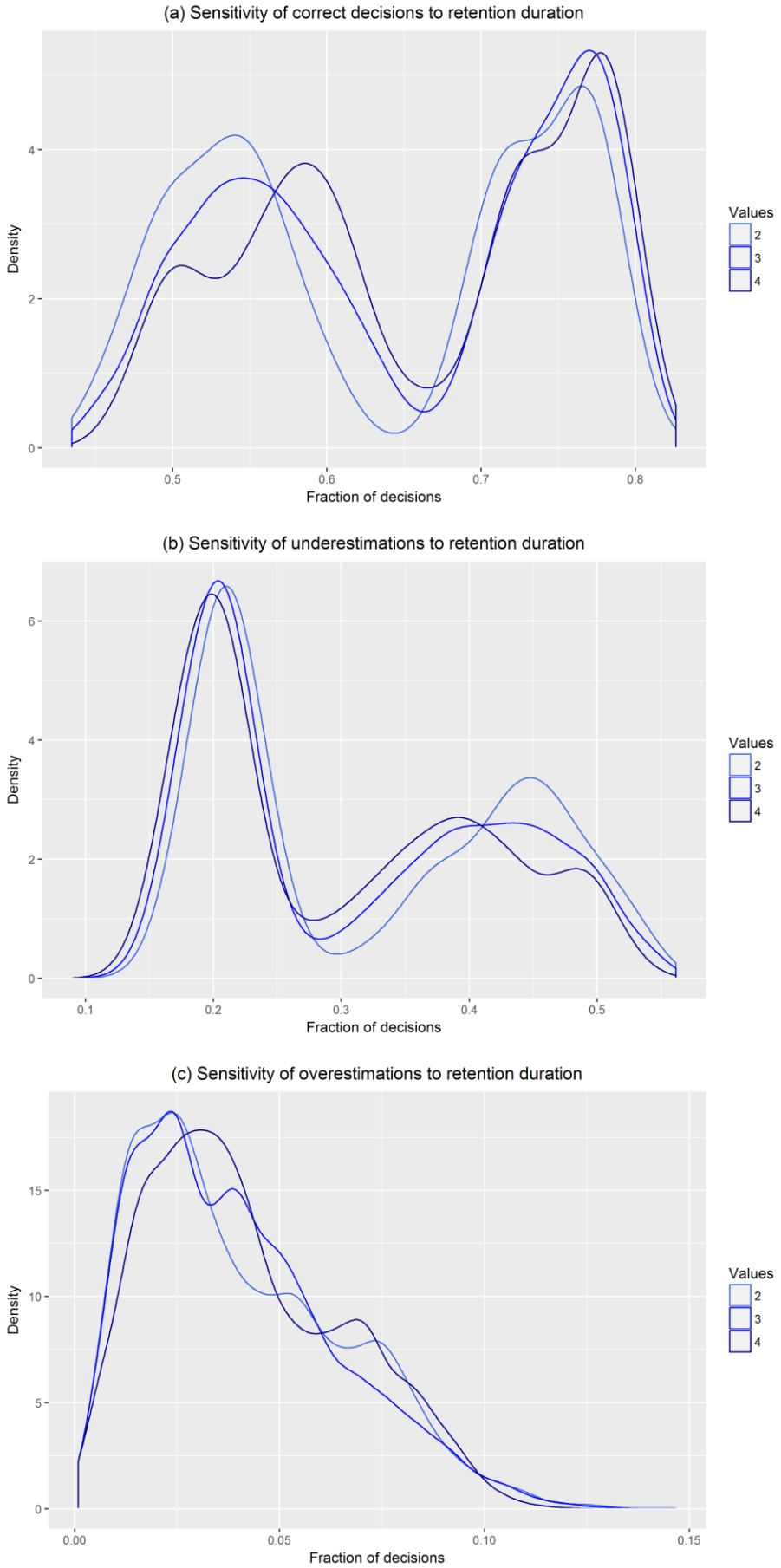


*Figure G-17: Effects of retention duration on decision correctness*

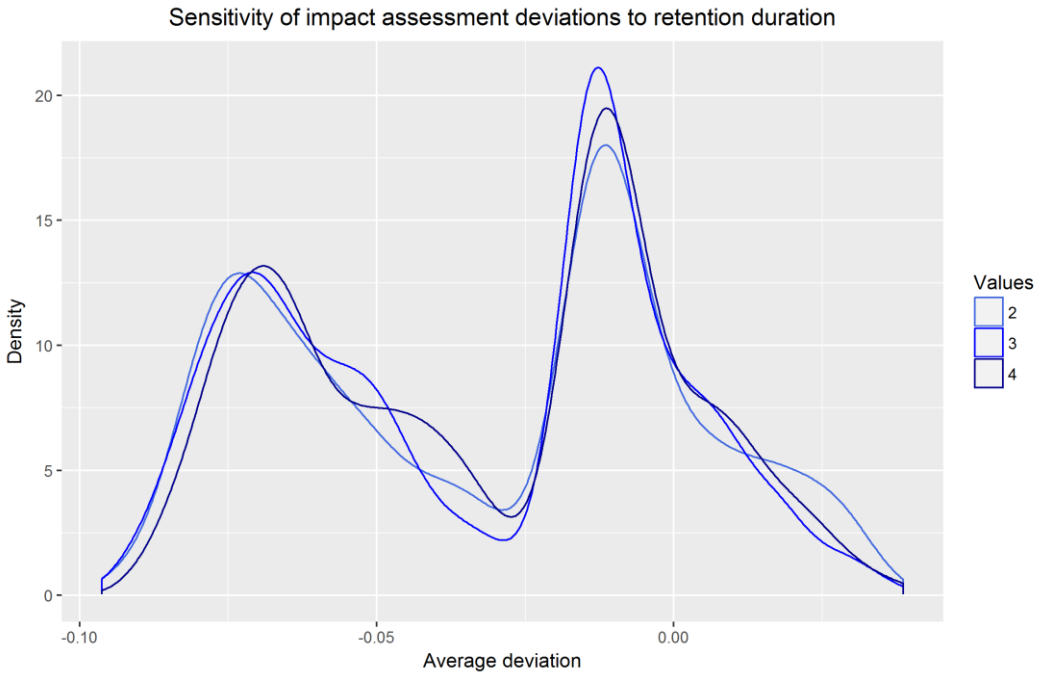


*Figure G-18: Effects of retention duration on impact assessment deviation*

## G.VII User traffic frequency

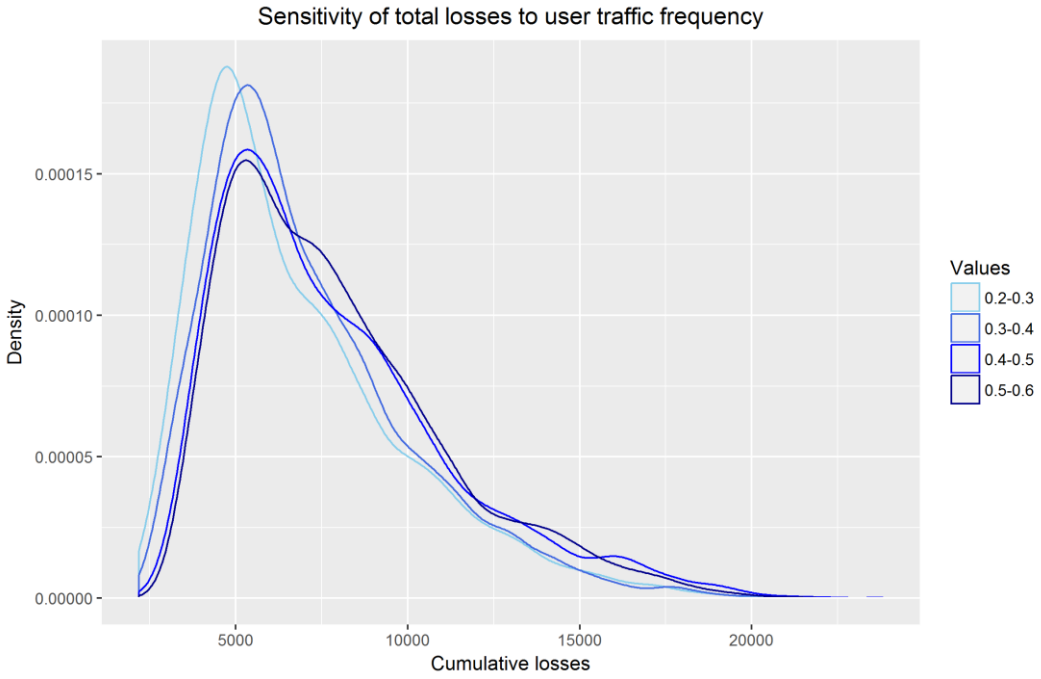


*Figure G-19: Effects of user traffic frequency on total losses*

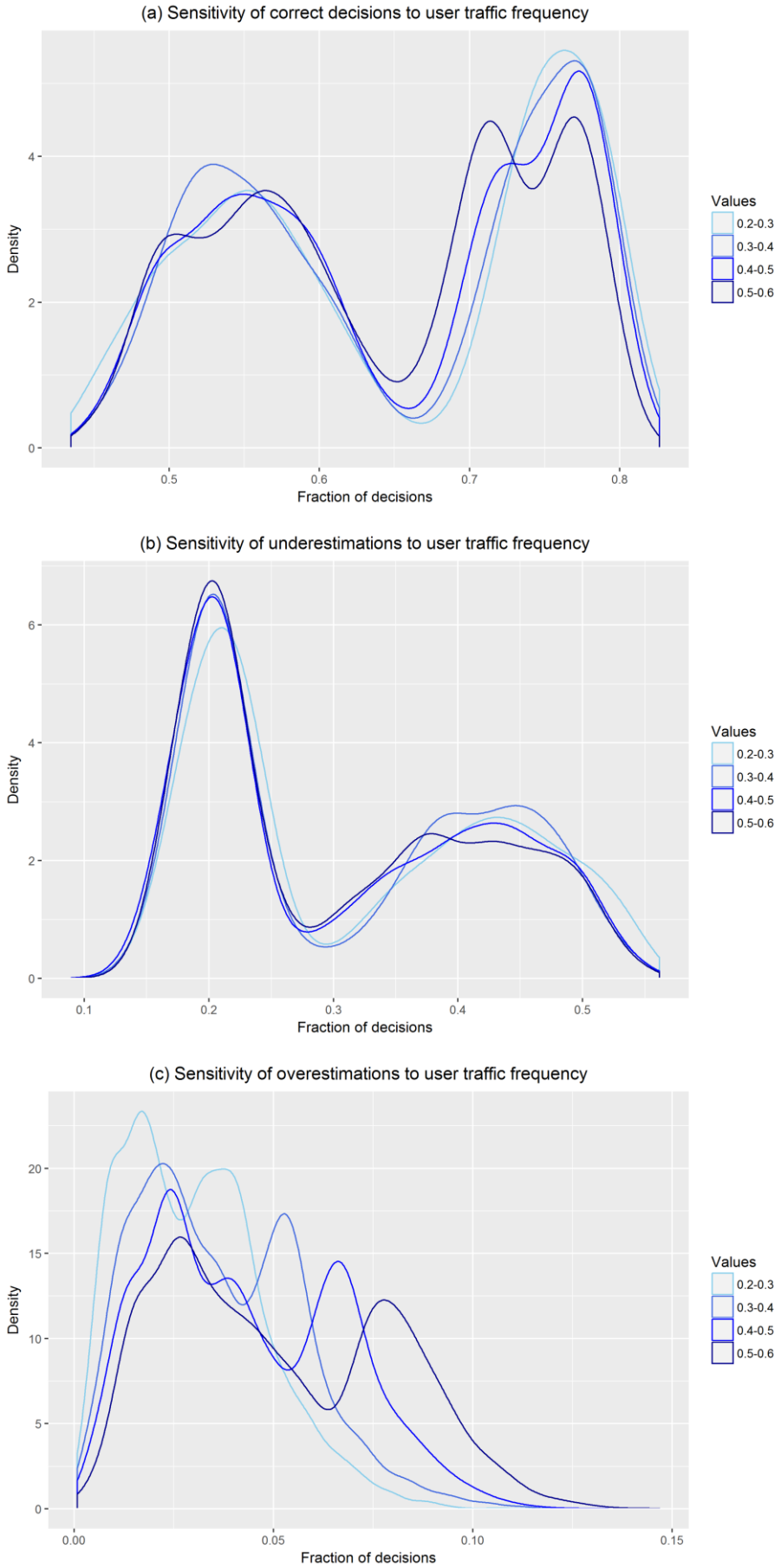


*Figure G-20: Effects of user traffic frequency on decision correctness*

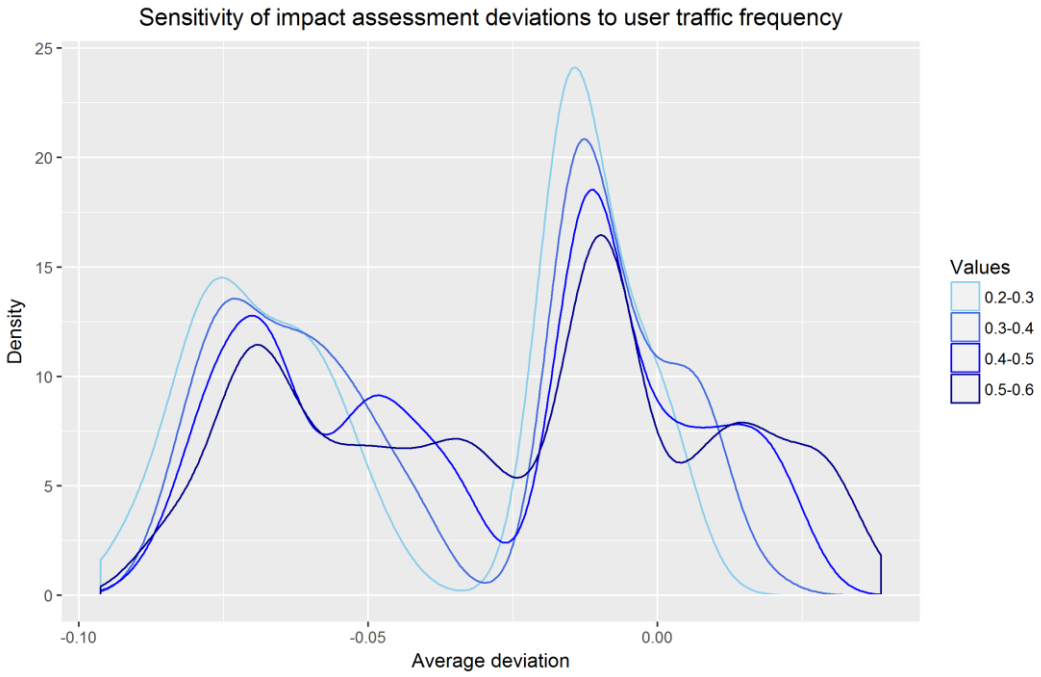


*Figure G-21: Effects of user traffic frequency on impact assessment deviation*

## G.VII User traffic criticality

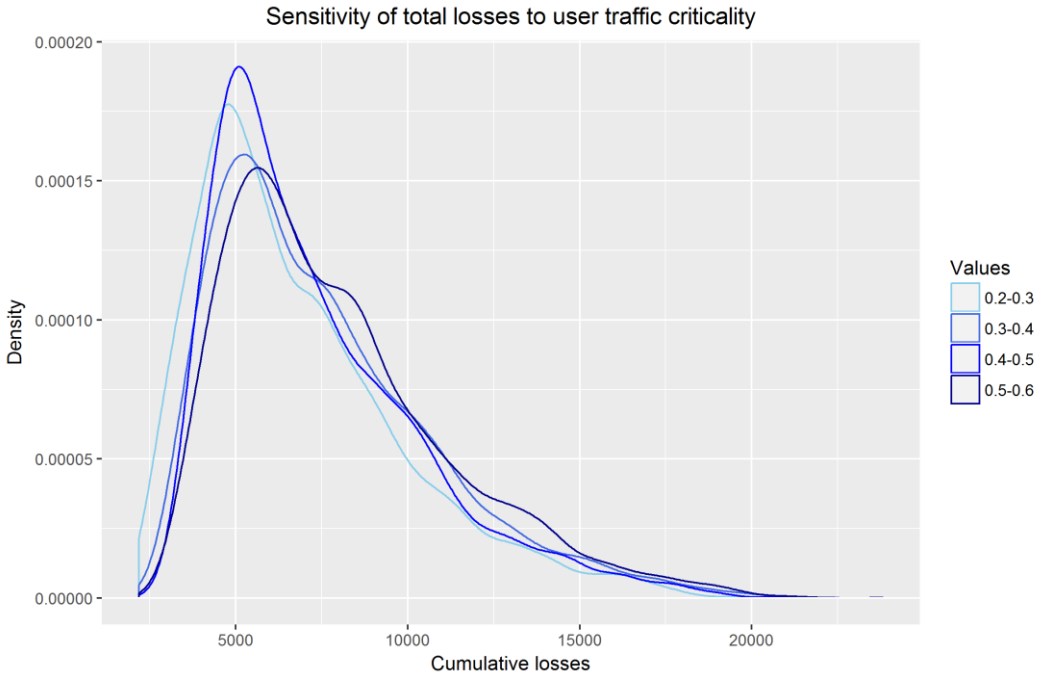


*Figure G-22: Effects of user traffic criticality on total losses*

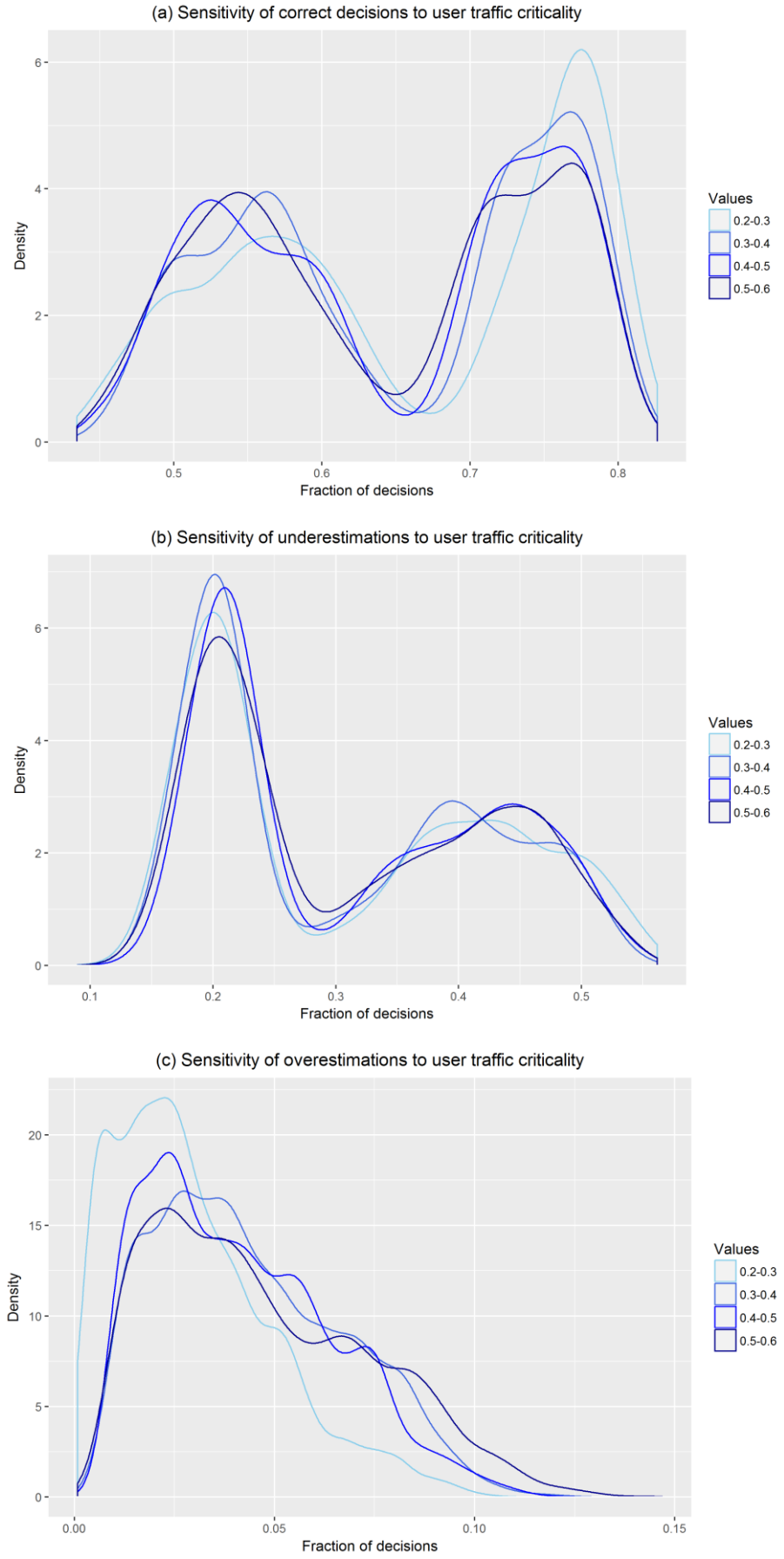


*Figure G-23: Effects of user traffic criticality on decision correctness*

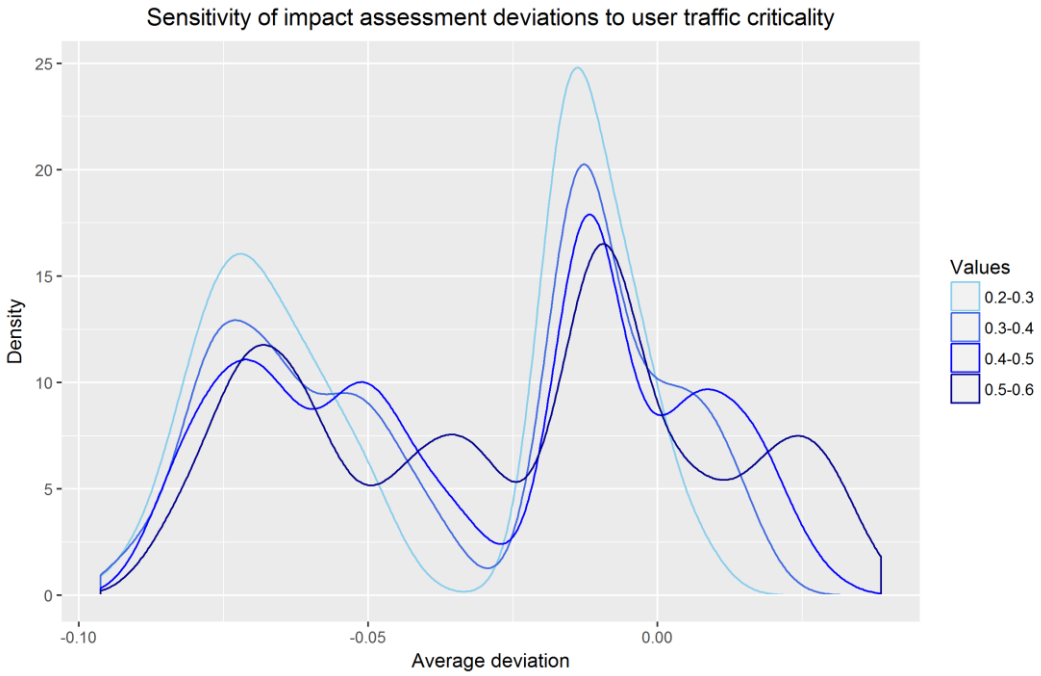


*Figure G-24: Effects of user traffic criticality on impact assessment deviation*