JYU DISSERTATIONS 407

Samir Puuska

Command and Control

Monitoring, Defending and Exploiting Critical Infrastructure

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY

JYU DISSERTATIONS 407

Samir Puuska

Command and Control

Monitoring, Defending and Exploiting Critical Infrastructure

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi elokuun 11. päivänä 2021 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
on August 11, 2021 at 12 o'clock noon.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2021

ABSTRACT

Puuska, Samir
Command and Control: Monitoring, defending and exploiting critical infrastructure
Jyväskylä: University of Jyväskylä, 2021, 50 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 407)
ISBN 978-951-39-8755-8 (PDF)

For securing critical infrastructure, this thesis aims to develop a common operating picture system, establish methods for detecting targeted cyberattacks, and investigate exploits against machine learning -based decision making. A design-science research framework is used, in which the validity is assessed through practical applicability of the solution artifact, and through an iterative requirements–evaluation cycle in close cooperation with key stakeholders.

The included studies address three topics: i) common operating picture systems, with emphasis on modeling and analysis methods, ii) neural network -based detection of encrypted malware command and control channels, and iii) one-pixel attacks targeting a neural network -based computer-aided cancer diagnosis. The studies made extensive use of raw data obtained through stakeholder collaboration. In addition, malware network traffic data generated through cyber-training activities on cyber-range environments, and tools used in targeted APT-malware attacks were utilized. A tissue sample -based tool, utilizing neural network technology, for computer-aided diagnosis of breast cancer, and associated digitized light microscope samples were used in vulnerability research.

The main results include ascertaining the applicability of the design-science research framework to the individual problem fields, and noting the necessity of raw data and stakeholder cooperation. Considering the results by topic, the required modeling and analysis methods could be implemented as a part of a common operating picture system, suitable neural network architectures with validation methods were created in malware traffic detection studies, and a method for producing hostile samples could be found in the study concerning one-pixel attacks.

The practical results of the common operating picture -study include an VN TEAS report, produced to support state-level decision making, in which the results of the studies were utilized extensively. With regard to cyberattack detection methods, their suitability for SUNBURST-backdoor detection was established. With regard to the one-pixel attack, the feasibility of the attack was demonstrated and the first publication considering the attack in a computer-aided diagnostic setting was produced.

Keywords: critical infrastructure protection, mathematical modeling, advanced persistent threat, intrusion detection, neural networks, one-pixel attack, computer-aided diagnosis

TIIVISTELMÄ

Puuska, Samir
Kriittinen infrastruktuuri: tilannekuva, puolustus ja vihamielinen vaikuttaminen
Jyväskylä: Jyväskylän yliopisto, 2021, 50 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 407)
ISBN 978-951-39-8755-8 (PDF)

Kriittisen infrastruktuurin turvaamiseksi pyritään kehittämään tilannekuvajärjestelmä, luomaan kohdistettujen verkkohyökkäyksien havainnontimenetelmiä sekä tutkimaan vihamielistä vaikuttamista koneoppimismenetelmäpohjaiseen päätöksentekoon. Tähän tarkoitukseen käytetään kehittämistutkimuksellista (design-science research) kehikkoa, jonka puitteissa validiteettiä arvioidaan sekä ratkaisuartefaktin käytännön soveltuvuuden, että iteratiivisen vaatimusmäärittely–evaluaatiosyklin kautta läheisessä yhteistyössä keskeisten sidosryhmien kanssa.

Osatutkimukset käsittelevät kolmea aihepiiriä: yhdistetyn tilannekuvan järjestelmää mallinnusja analyysimenetelmineen, haittaohjelmien salattujen komentokanavien neuroverkkopohjaista paljastamista sekä vihamielistä yhden kuvapisteen erheytyshyökkäystä neuroverkkopohjaiseen syövän tietokoneavusteisen diagnoosin työkaluun. Osatutkimuksissa hyödynnettiin laajasti sidosryhmäyhteistyön kautta hankittua raakadataa, kyberharjoitustoiminnan ja -ympäristön avulla tuotettua haittaohjelmien verkkoliikennedataa, kohdistetuissa APT-ryhmien haittaohjelmahyökkäyksissä käytettyjä kyberoperaatiotyökaluja sekä kudosnäytepohjaista rintasyövän tietokoneavusteisen diagnoosin neuroverkkoteknologiaa hyödyntävää työkalua ja digitalisoituja valomikroskooppinäytteitä.

Tutkimuksen päätuloksina voidaan osaltaan pitää valitun kehikon sovelluskelpoisuutta osatutkimusten ongelmakenttiin, sekä tutkimusten osoittamaa raakadatan ja sidosryhmäyhteistyön välttämättömyyttä. Tilannekuvajärjestelmän osatutkimuksissa kyettiin toteuttamaan vaaditut mallinnus- ja analyysimenetelmät, havainnointimenetelmien osuudessa luotiin soveltuvat neuroverkkoarkkitehtuurit validointimenetelmineen sekä erheytyksen osatutkimuksessa löytämään menetelmä vihamielisten näytteiden tuottamiseksi.

Tutkimuksen käytännöllisinä tuloksina voidaan tilannekuvajärjestelmän osalta pitää valtiollisen päätöksenteon tueksi tuotettua VN TEAS -raporttia, jossa osatutkimusten tuloksia hyödynnettiin laajasti. Verkkohyökkäyksien havainnointimenetelmien osalta voidaan todeta niiden soveltuvuus SUNBURST-takaoven havainnointiin. Erheytyshyökkäyksen osalta voidaan tuloksiksi lukea käyttökelpoisuuden osoitus sekä aiemmin julkaisematon kuvaus hyökkäystyypin kohdistamisesta tietokoneavusteisen diagnoosin sovellutuksiin.
Avainsanat: kriittinen infrastruktuuri, matemaattinen mallinus, APT-uhka, kyberhyökkäysten havaitseminen, neuroverkot, yhden pikselin hyökkäys, tietokoneavusteinen diagnoosi

Author Samir Puuska
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors Professor Timo Hämäläinen
Faculty of Information Technology
University of Jyväskylä
Finland

Adjunct Professor Tero Kokkonen
Institute of Information Technology
JAMK University of Applied Sciences
Finland

Reviewers Associate Professor Mika Ylianttila
Faculty of Information Technology and Electrical Engineering
University of Oulu
Finland

Professor Mohammed Elmusrati
School of Technology and Innovations
University of Vaasa
Finland

Opponent Professor Kimmo Halunen
Faculty of Information Technology and Electrical Engineering
University of Oulu
Finland

ACKNOWLEDGMENTS

This project would not have been possible without financial support from the Finnish Funding Agency for Technology and Innovation (TEKES), the Finnish Prime Minister's office (VN TEAS), the Scientific Advisory Board for Defence (MATINE), the European Union's framework programme Horizon 2020, the Regional Council of Central Finland, Council of Tampere Region, and the European Regional Development Fund.

I would like to extend my deepest gratitude to my supervisors, professor Timo Hämäläinen and adjunct professor Tero Kokkonen, for their advice, guidance, and support. I would also like to thank the pre-examiners, professor Mohammed Elmusrati and associate professor Mika Yliant-tila, for their insightful comments. I would also like to express my deepest appreciation to all of my co-authors and collaborators. I have had the fortune of participating in the work of many awesome research groups and projects. Thank you all!

I also wish to thank the JYU Faculty of Information Technology, JAMK Institute of Information Technology, JYVSECTEC, the Department of Military Technology of the National Defence University (FIN), and the VTT Technical Research Centre of Finland for giving me the opportunity work on this dissertation.

Finally, I would like to thank all of my family, friends, and my cats for all the forms of support too numerous to mention!

Helsinki
July 19, 2021

Samir Puuska

CONTENTS

LIST OF INCLUDED ARTICLES

P1 S. Puuska *et al.*, "Modelling and real-time analysis of critical infrastructure using discrete event systems on graphs", in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2015, pp. 1–5. DOI: 10.1109/THS.2015.7225330

P2 S. Puuska *et al.*, "Integrated platform for critical infrastructure analysis and common operating picture solutions", in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2017, pp. 1–6. DOI: 10.1109/THS.2017.8093737

P3 S. Puuska *et al.*, "Nationwide critical infrastructure monitoring using a common operating picture framework", *International Journal of Critical Infrastructure Protection*, vol. 20, pp. 28–47, 2018, ISSN: 1874-5482. DOI: 10.1016/j.ijcip.2017.11.005

P4 T. Kokkonen and S. Puuska, "Blue Team Communication and Reporting for Enhancing Situational Awareness from White Team Perspective in Cyber Security Exercises", in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, O. Galinina *et al.*, Eds., Cham: Springer International Publishing, 2018, pp. 277–288, ISBN: 978-3-030-01168-0. DOI: 10.1007/978-3-030-01168-0_26

P5 S. Puuska *et al.*, "Anomaly-Based Network Intrusion Detection Using Wavelets and Adversarial Autoencoders", in *Innovative Security Solutions for Information Technology and Communications*, J.-L. Lanet and C. Toma, Eds., Cham: Springer International Publishing, 2019, pp. 234–246, ISBN: 978-3-030-12942-2. DOI: 10.1007/978-3-030-12942-2_18

P6 T. Kokkonen *et al.*, "Network Anomaly Detection Based on WaveNet", in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, O. Galinina *et al.*, Eds., Cham: Springer International Publishing, 2019, pp. 424–433, ISBN: 978-3-030-30859-9. DOI: 10.1007/978-3-030-30859-9_36

P7 S. Puuska *et al.*, "Statistical Evaluation of Artificial Intelligence -Based Intrusion Detection System", in *Trends and Innovations in Information Systems and Technologies*, Á. Rocha *et al.*, Eds., Cham: Springer International Publishing, 2020, pp. 464–470, ISBN: 978-3-030-45691-7. DOI: 10.1007/978-3-030-45691-7_43

P8 T. Sipola *et al.*, "Model Fooling Attacks Against Medical Imaging: A Short Survey", *Information & Security: An International Journal*, vol. 46, no. 2, pp. 215–224, 2020. DOI: 10.11610/isij.4615

P9 J. Korpihalkola *et al.*, "One-pixel Attack Deceives Automatic Detection of Breast Cancer", *Computers & Security (under review)*, 2020. eprint: arXiv:2012.00517

1 INTRODUCTION

The modern world is dependent on ubiquitous availability of computing resources. This demand arises from virtually any industrialized human activity, which requires vast computational powers to operate on the global scale. Even social activity and normal human interactions are now intertwined with computational platforms that facilitate communication, analyze behavior, and alter social and physical environments. As these technologies have been advancing, so has our reliance on them. Automation, in various forms, now controls the most essential systems responsible for vital societal functions.

Critical infrastructure, the systems that form the basis structure for vital societal functions [17, 48], is evolving and growing. In the future, critical infrastructure will encompass ever-increasing number of technological solutions humans have created as answers to questions like global communication, food security, and climate change [62]. Sometimes this development is fast: in a relatively short time span the COVID-19 pandemic has created a world where telecommuting could become the "new normal" [10]. It is no wonder, then, that understanding the nature of this formidable environment, protecting it from threats, and understanding its weaknesses are essential. Even though old threats seem to never die, the modern digital ecosystem has created new ways for malicious endeavors [28]. It is no longer enough to understand ordinary faults that all systems develop, we now have to actively defend ourselves against, at times, well-resourced and determined adversaries [9]. The ever-increasing complexity and expanding threat landscape compel us to research and develop solutions that allow us to monitor, defend, and understand exploits against critical infrastructure, which automation now controls [62].

Cybersecurity and critical infrastructure protection are vast fields. Although they have recently received much attention both in and outside of academic circles, the complexity of the modern cyberphysical world has perhaps more gaps than well-researched areas. This is especially true for viewpoints considering various attacks and attack surfaces. There are factors that materially complicate research of the cyber domain and critical infrastructure. Gaining access to data and experts is difficult. Further complications arise due to the open nature of science, which proves to be problematic when dealing with potentially sensitive details on critical infrastructure, or complex cyberattacks against systems in production.

The threat, and increasingly the potential, of the modern cyber environment has not gone unnoticed at the nation-state level. Intelligence agencies, military organizations, and other groups

around the world have been developing and using their cyber capabilities, in sometimes plainly visible ways, for conducting their operations. This trend is likely to continue [61]. At the same time, many countries and organizations have found out that their ability to withstand cyber attacks leaves much to be desired. Advanced adversaries do not necessarily benefit from scientific research on critical infrastructure exploitation, as they are independently resourced for discovering that capability. By addressing attacks and their mitigation in scientific research, the defending organizations and general public gain understanding of what they are facing, and have a chance of detecting and foiling these attacks.

The increasing amount of data the modern world produces has long since eclipsed the natural human capability for processing it. Instead, we have created technologies that can do processing, analysis, and even decision making for us. Raw data is useless without a way to interpret the numbers and characters in a context that allows us to benefit from them. Statistical inference as a method for problem solving is not new; historical records show examples of this centuries before the Common Era. What has changed is the scale on which we can collect raw data and perform these calculations. Along the centuries we also have devised new methods and refined old ones in furthering the endeavor to achieve human-like thinking using machines. Naturally, these solutions have found their way into cybersecurity and critical infrastructure protection. The role of artificial intelligence and machine learning in these fields is complex. On one hand, they can be used to detect many forms of misuse ranging from financial fraud to network intrusions. On the other hand, they are increasingly used to mount advanced attacks against both automated systems and humans [104].

1.1 Research questions and methodology

The aim of this thesis is to consider critical infrastructure from several viewpoints, rather than focus on one narrow section. This thesis and the included articles address critical infrastructure from three different thematic categories: monitoring, defending, and exploitation. Figure 1 illustrates how the included scientific publications are grouped into the categories and sub-categories of each theme. The first theme explores challenges in monitoring critical infrastructure, and means for processing and presenting data in a fashion that allows a human operator to make inferences on the current and future state of the infrastructure as a whole. The second theme explores the role of *artificial intelligence (AI)* and artificial neural networks in detecting advanced malware attacks often directed against computer networks vital for the operation of critical infrastructure. The third theme explores healthcare, a critical infrastructure field currently enjoying increases in AI automation, from the viewpoint of exploitation.

Each of the themes and the corresponding publications have their own sets of specialized research questions. Despite their differing viewpoints, there are certain high-level questions that are shared between the three categories.

1. From one of the viewpoints, what salient problems does critical infrastructure have?
2. What are the real-life requirements for a suitable solution?
3. How do we acquire raw data from real systems?
4. How can we construct a functional prototype artifact?
5. Does the constructed prototype achieve the required real-life effect or performance?

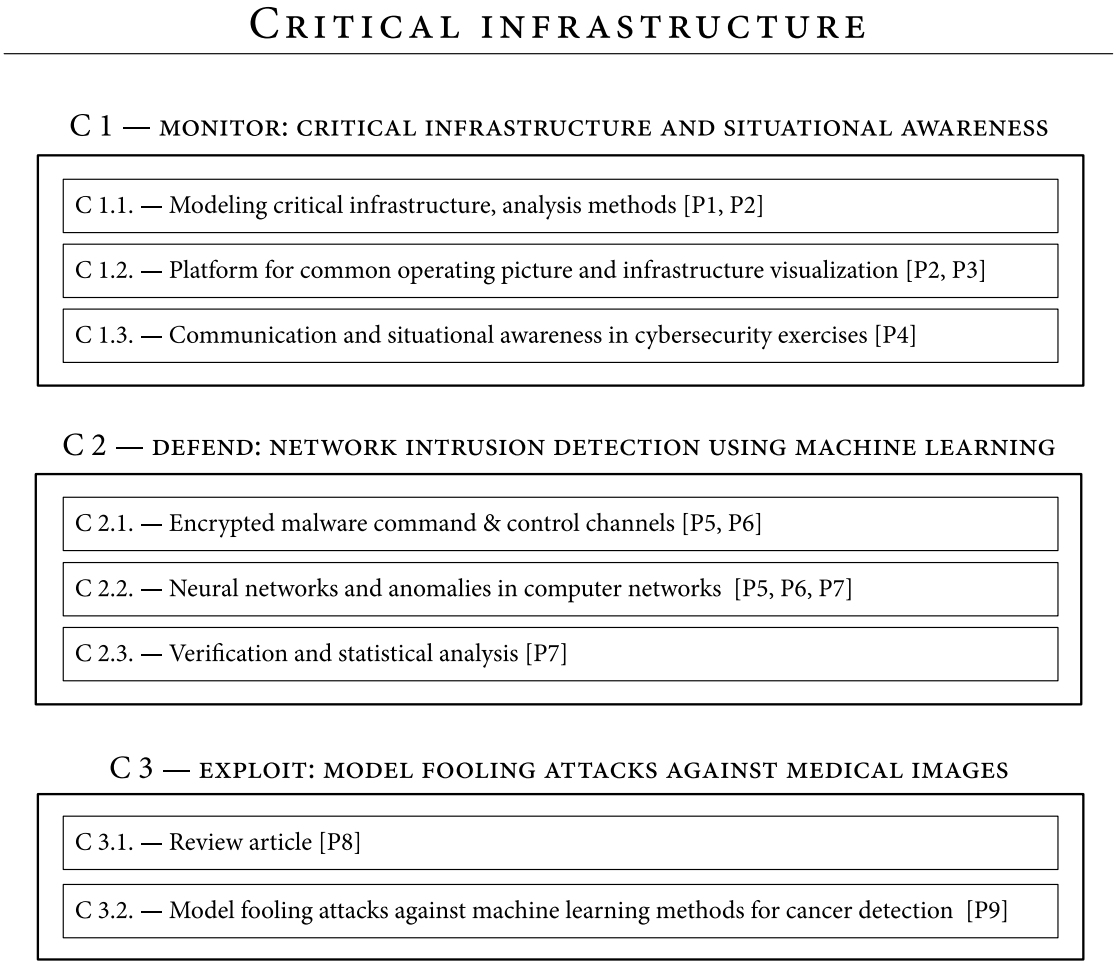


Figure 1: The three thematic categories and their sub-categories addressed in this thesis. Square brackets indicate papers that include elements from respective sub-topics.

No research should be an island. The work in this thesis was carried out as part of several larger research projects. This is also reflected in the framing of each individual paper's goals and focus, as the exact requirements are often products of prior research conducted by other members of the research team, or are otherwise not a part of this thesis.

In applied research, the end target is to create solutions that have a high chance of working under real-life situations. To this end, the research methodology and the methods must reflect this goal [60]. The solution tends naturally towards producing a prototype, as that prototype can then be iteratively improved for example via user testing, field experimentation, or various collaborative means. This sort of approach is known as *design-science research (DSR)*, or alternatively as *constructive research* methodology [26, 47]. DSR is a solution-focused, participatory, and iterative methodology, as opposed to the more observational and problem-focused approaches associated with traditional science [11]. Design science is focused on the artificial, and DSR is a methodology that produces *artifacts*, i.e. artificial things that are synthesized by human beings, and discussed in terms of functions or goals [92]. A prototype, as an artifact, creates means for exploration of the problem, development, and finally evaluation of the proposed solution [11, 70]. Traditional statistical tests, trials, and other such methods are used in conjunction with iterative processes that take into account how stakeholders, the intended users of the results, see the proposed solutions and their viability. The stakeholders may even be the original proposers of the main problem, which is then formulated as a series of research questions by the research team. This iterative approach, when successful, extends the validity of the research beyond what traditional statistical tests and research designs could provide. Ideally, there is then just a short leap into operationalization to production. The DSR methodology relies heavily on using data and subject-matter experts to drive design and in selecting the requirements [11]. All the papers included in this thesis rely on expert interviews, user tests, raw data produced by real systems, or a combination thereof. The central theoretical foundation and main challenges of each three thematic categories are presented in Chapter 2. The detailed account of aims, data, methods and results of each publication are presented in Chapter 3. Discussion of the results and conclusions are presented in Chapter 4.

1.2 Publications and author's contribution

The author's contribution to the included articles varies. Paper **P1**: The author is responsible for the original idea for the proposed model, gathering and collecting the test data, formalizing the model, as well as being the main writer of the article. Paper **P2**: The author is responsible for developing the idea and major parts of the software for the proposed simulator and middleware, in conjunction with the other authors. The author further participated in gathering, analyzing, and refining the data required for running the simulations. The author developed the geographic information system view, for visualization in the COP system. All named authors participated in the writing process. Article **P3**: The author is responsible for designing and developing the data collection middleware solution, the analysis methods, and some of the server-side user interface code. The author also chiefly participated in statistical analyses, as well as contributed most of the article's text. This article has appeared as a part of another dissertation, without overlapping contribution between authors [102]. Paper **P4**: The author is responsible for developing the idea

and concept, as well as creating the reporting tool and for collecting and analyzing the data. The author also contributed text to the article, in conjunction with the other authors. Paper **P5**: The author contributed the central concept, and participated in data collection, analysis study design, as well as writing. Paper **P6**: The author contributed to the overall design of the study, feature engineering and evaluation, data collection and analysis, and writing. Paper **P7**: The author contributed the main concept, study design and most of the text, as well as participated in selecting suitable statistical methods and distributions. Paper **P8**: The author contributed to the literary review and writing. Paper **P9**: The author is responsible for conceptualization, methodology, data processing, software, and participated in writing the original draft.

2 THEORETICAL FOUNDATION

In this chapter the relevant theoretical foundations are presented in brief detail. The chapter does not attempt to address these subjects comprehensively: The goal is to present central concepts, case examples, and challenges in these relatively disjoint topics, enabling the reader to consider the included articles in context.

2.1 Critical infrastructure and situational awareness

Critical infrastructure (CI) refers to systems that form the basis structure for vital societal functions [48]. The European Council, for example, highlights health, safety, security, economy, and social well-being as examples of functions that should be considered vital [17].

2.1.1 *Common operating picture*

Critical infrastructure is a complex environment, with complex relationships. The task of maintaining *situational awareness (SA)* about the state is one of the prominent research areas of the field [14]. By definition, CI is critical, and there is massive incentive to holistically monitor its functionality, and predict the extent and impact of current and future failures in real time. Both governmental and private-sector actors are interested in monitoring their own assets, as well as the state of other systems they are dependent on. In order to effectively disseminate and utilize information, each actor is required to share details of their system in a controlled way. This sharing can be incentivized by making information sharing mutually beneficial [103].

A platform to share information, along with suitable analysis functionality and visualization techniques provide a so-called *common operating picture (COP)* solution. Although military in origin, COP in CI context refers to a platform where all the sectors are represented together using data fusion and visualization tools [103]. CI spans every infrastructure sector, and the breadth of devices and systems that must be integrated grows large. Some systems, such as those connected directly to the Internet, are very easy to monitor remotely by their nature, others may require a human-in-the-loop approach. Research areas include data collection and fusion elements, a task complicated by the diversity of CI components [48].

The analysis capability of a COP system is tied to the task of maintaining the situational awareness of human operators. As proposed by Endsley, SA includes three levels of comprehension, consisting of understanding current elements, their relation to each other, and the future developments of the system as a whole [14]. Consequently, the analysis capability should provide suitable information on each of the SA levels in a way that assists the operator in maintaining SA. As maintaining SA is an ongoing effort, the underlying model must be capable of operating in real time, and provide continuous output and forecasts as the situation evolves, while tolerating disruptions in data delivery.

2.1.2 *Modeling interdependencies, predicting cascading failures*

One of the challenges associated with CI is recognizing what and where those critical assets are [48]. When this work was first conducted in 1990s, it was swiftly discovered that the infrastructure was highly interconnected: both physically, and via telecommunication systems. Latter research went on to call CI as *interdependent* [48]. Rinaldi *et al.* define interdependent as "highly interconnected and mutually dependent in complex ways", as it was discovered that failures on one part of CI may cause *cascading failures* impacting other parts of CI [80]. CI is often owned and controlled by various public and private parties, further complicating the relationship between its various parts.

Much of the research on CI is focused on studying the interdependencies. This field encompasses researching suitable mathematical and technical models, and mapping and observing CI structure and events as they appear in the real world. Both of these research activities are somewhat hindered by the sensitive nature of these systems, as well as the fragmented ownership landscape. There is also a conflict between the open nature of scientific research and publishing, and the sensitive nature of CI datasets.

Various different modeling approaches have been proposed in academic literature [66]. One of the particular challenges in creating a CI model for a COP system is keeping the individual model relatively simple, allowing the chaining of the modeled components and influences to simulate the interdependent nature of CI at scale. The model should also provide some estimates on how severe an observed failure was, and how it relates to the systems that are dependent on its operation. Systems like cellular base-stations are dependent on external power, although they may operate using emergency battery power for several hours. This creates a time-sensitive component to the model. A COP system receives status updates from some of the infrastructure components periodically. The model should both use these updates to keep up to date, as well as interpret the cessation of these updates as a sign of failure. Papers in C 1.1 describe a model based on graphs and finite state transducers, and then present an application of that model to a real-world use case.

2.2 Computers, networks, and intrusions

Computers today are rarely used without a network of some kind. This state of affairs brings innumerable advantages, but in addition it also makes it easier for attackers to operate clandestinely, as the amount of traffic is too vast for humans to manually inspect, and encryption has become virtually ubiquitous. We firstly present a motivating example of an attack, where the

methods presented in this thesis would likely have been effective in mitigating the impact. A short introduction to the basic concepts of neural networks is given, followed by an overview of network intrusion detection using this type of machine learning approach. Finally, some remarks concerning the statistical side of the phenomenon are discussed.

2.2.1 *SUNBURST: a tool for global espionage*

On December 13, 2020, American cybersecurity company FireEye Inc. published details on how an advanced nation-state -sponsored attacker had compromised numerous high-value targets using a so-called supply-chain attack [20]. The attacker had installed a malicious backdoor into a widely used network and infrastructure management platform Orion, developed by SolarWinds Inc. [8, 94]. Using this trojanized software, the allegedly Russian *advanced persistent threat (APT)* group gained access into numerous systems where the management platform was deployed, including several used by the United States federal government [9, 61].[1] Various cybersecurity companies, including FireEye, refer to the malicious code as SUNBURST [20]. SUNBURST attempted to conceal many of the malicious connections by mimicking a legitimate update process. This approach proved to be successful, and SUNBURST was only detected when the attacker had already used it to exfiltrate documents and other data from the systems.

SUNBURST is the first part of an attack chain. By using multiple stages, the attacker can target high-value organizations via customized payloads. In several documented cases, SUNBURST was used to deliver a malware dropper known as TEARDROP. The purpose of TEARDROP is to deploy yet another payload, a modified Cobalt Strike BEACON [56]. Cobalt Strike is a tool suite for cyber adversary simulation, developed by Strategic Cyber LLC.[2] It has the same capabilities as advanced malware, and is therefore used in malicious attacks in addition to legitimate use by red teams. Cobalt Strike BEACON is among the malware samples used in articles **P5** and **P6**.

2.2.2 *A short introduction to neural networks*

The term "machine learning" was first used in 1959 [87]. Since then, the field has seen the era of big data and, with it, incredible increase of computational performance. A common problem in machine learning is to construct a function based on some limited set of example data. The goal is for the function to generalize from the training examples in such a way that other data also performs desirably. For example, if one has a set of cat pictures, a machine learning method could be used to create a function that can recognize if cats appear in other pictures as well, perhaps the instant a user takes one with a smartphone [43].

Artificial neural networks (ANN) and so-called deep learning have become household names during the last few years, and are known for their apparent applicability to big data problems. However, artificial neural networks are not new; surprisingly, the concept predates the term "machine learning". ANNs are often represented as a kind of a digital counterpart to biological cell-

[1] The SUNBURST situation is ongoing, and new details are constantly emerging. As the event progresses, this section may no longer contain the most current information.

[2] https://www.cobaltstrike.com

based brains [82].[3] They have the ability to generalize a function from a finite set of training examples, without needing extensive human input to guide the process. This also means that there is no fundamental requirement to understand intricate theory and mathematics behind the method, or the phenomenon under study, before using ANNs; the field relies, quite strongly, on empirical results showing the method ostensibly working, while theoretical guarantees and understanding lag behind the cutting edge applied research.[4] This has not prevented the field from achieving major successes.

The so-called *supervised learning* considers how *labeled training data* can be used to construct a generalized function that *predicts* the label for other similar data as well. Consider $F : \mathbb{R}^n \to \mathbb{R}^m$, a function that maps a vector from $\mathbb{R}^n$ to a vector in $\mathbb{R}^m$. We can use this rather abstract notation to present a problem: if we have a set T of ordered pairs (n, m), where $n \in \mathbb{R}^n$ and $m \in \mathbb{R}^m$, can we construct a function which returns desirable results for some of the points in $\mathbb{R}^n$, even though they did not appear in the training set T? We use inexact terms like "desirable" and "for some of the points" here for a reason. In machine learning we often lack a way of expressing certain subsets of $\mathbb{R}^n$ in mathematical form. Conceivably, we can represent a digital picture of a cat as a vector in $\mathbb{R}^n$, but we immediately run into a problem if we try to mathematically define what subset of $\mathbb{R}^n$ are the vectors containing a cat picture. The output $m \in \mathbb{R}^m$ can be defined as a binary, one or zero, depending on if the input is a cat picture or not. Even with this mathematically ill-defined problem, it is possible to use ANNs to detect cats, if given a sufficient amount of training data [43].

The mechanics of artificial neural networks are ruled by expedience; they have mathematical properties that make them sufficiently universal, as well as numerically tractable. Artificial neural networks, in essence, leverage a simple non-linear function, applied repeatedly, to approximate other functions [25]. The non-linearity causes ANNs to be *universal approximators*, allowing them to represent a wide class of functions [29]. The (sigmoid) logistic function,

$$\sigma(x) = (1 + e^{-x})^{-1} \tag{1}$$

is an example of such non-linear function [7]. It should be noted, however, that it is not by any means the only suitable choice [40]. This non-linear *activation function* is so named to reflect the terminology used when describing similar behavior in biological neurons. The activation function does not have any adjustable parameters. Parameters are needed to "fit" the non-linear function to the function that we are trying to approximate. For that purpose, we introduce two parameters for scaling and shifting the input, called *weight* (W) and *bias* (b) in ANN parlance. The parameters are applied before the activation function, yielding the form $\sigma(Wx + b)$. This construct is known as a *neuron*, again a reflection of the nomenclature used in biology.

One neuron does not a neural network make. To approximate complex functions, the neurons are usually set up in layers, forming a network. One of the more common configurations is a *fully connected network*, where the output of each neuron in a layer is passed to every neuron on the next layer, the first layer acting as input, and the last as output. We now introduce a more concrete definition for fully connected networks, bringing us closer to the actual numerical approach. We

[3] While this analogy is useful in a limited way, it also misleadingly suggests that the networks of artificial "cells" share similar traits comparable to biological brains and their capabilities.

[4] This philosophy is reflected in this chapter, where some of the mathematical rigor and nuance is sacrificed for readability and brevity.

expand the definition of function (1) to cover vectors in component-wise fashion; if x is a vector, the function is applied to every component separately. $\sigma(Wx + b)$ can now be understood as the operation on a single layer, where W is now a matrix, and b a vector [27]. The dimensions of weight matrix W are defined by the number of neurons at the previous layer (columns), and the number of neurons at the current layer (rows). Bias vector b matches the number of neurons at the current layer. Combined, the weight matrices and bias vectors for each layer constitute the *parameters*, θ, of the network. It is now possible to see the repeated application of the non-linear function, for example in the case of a three-layer network

$$F(x) = \sigma\Big(W_3\,\sigma\big(W_2\,\sigma(W_1 x + b_1) + b_2\big) + b_3\Big)$$

of unspecified dimensions.

The goal is to learn "good" parameters for the ANN using a set of training examples. Continuing our example, we have a set of input points in $\mathbb{R}^n$, and corresponding target output points in $\mathbb{R}^m$. We now need to adjust the parameters of a network to produce the desired output in $\mathbb{R}^m$, when given a training example $\mathbb{R}^n$, for every training example in the set. There are, to be sure, multiple ways to achieve this goal. However, currently the most popular family of methods are *gradient*-based. Gradient-based optimization methods require the use of a *cost function*, which is a type of objective function that is minimized in a process called *training*. There is a choice of cost functions that are suitable to use with gradient methods. As an example, consider the well-known *quadratic cost function*,

$$C_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \left\| y(x_i) - F_\theta(x_i) \right\|_2^2 \tag{2}$$

also known as mean squared error (MSE) [5], with F_θ being the parametrized function representing the ANN model.

Local search optimization algorithms, such as gradient methods, became viable only when computers became powerful enough to perform the necessary calculations effectively, at scale. MSE itself predates that time, as do many other methods that use it, again illustrating that central ideas fueling ANNs span several centuries [46]. Local search works by using the objective function to measure how "good" the current state is, and then using some means to move to another solution, until a sufficiently optimal state is found. In other words, the *gradient descent* method iteratively minimizes the cost function. As the name implies, the method uses partial derivatives to guess how parameters should be altered to reduce the cost of the next state [6, 44].

Modern artificial neural networks tend to be large, some language models surpassing 10 billion parameters [79]. Naive gradient descent requires repeated calculation of the derivative of the cost function. Unfortunately, the closed-form solutions for derivatives to already massive $F(x)$ would be intractable, even with today's computing power. Instead, we use a set of practices that massively reduce the amount of computations via slight trade-offs to the optimality of the result. These methods include Taylor approximations, stochastic sample selection, and automatic differentiation [45, 49, 53, 81, 83]. The cost function is minimized, until a stopping criteria is reached.

In cases where the input vector is known to be structured in a certain way, it is possible to create an ANN capable of using that information. The canonical example of such structure are

images, where pixels are usually related to the ones next to them, creating patterns, such as cats. Patterns like these are almost exclusively what ANNs are supposed to detect and classify, no matter where in the picture they are. A *convolutional ANN* is a specialized network architecture that can exploit these local dependencies between pixels, while having much fewer parameters than a fully connected network would have [41]. This position independence is useful in many other tasks, such as detecting patterns in time-series. The convolutional *receptive field* can be thought as a form of *regularization*, controlling the bias-variance trade-off by limiting the set of functions the ANN is likely to learn [88].

2.2.3 *Intrusion detection: finding network anomalies*

Intrusion detection refers to the activity and technologies intended to identify various intrusions against computers and networks [73]. *Intrusion detection systems (IDS)* are purposefully built analysis tools which detect malicious events or activity, and report the "intrusion" for further analysis. The methods employed by a particular IDS depends on what is the nature of intrusion the system is set up to detect. For example, an IDS used to detect intrusions at the network level may use captured network traffic in their analysis. Since these so-called network intrusion detection systems are not able to see or control the software at the endpoints, they are unable to perform certain tasks the endpoints can. They cannot communicate with either endpoints, or alter the data being sent between them.

Cyber attacks come in many forms. The exact approaches: tactics, techniques and procedures (TTPs), are determined by the goals of the attacker, as well as their skill level. In many cases the attacker wants to gain access to information stored on various systems, as opposed to destroying or maliciously altering the records. Cyber attacks often have multiple phases, and require the attacker to actively control the malicious programs on the target systems. This requires a *command and control (C&C)* channel, a covert way for the attacker to relay instructions and receive data back from compromised systems. Naturally, ubiquitous encryption has not escaped malware authors. Many malicious C&C channels attempt to hide amongst legitimate web traffic by mimicking normal browsing, to varying levels of success.

The traffic computers generate when communicating with each other through networks is, in a sense, very varied. The applications people use every day range from video games and web-based social media to suites such as the Microsoft Office and Outlook, to give examples from this diverse set. On the other hand, the traffic of these varied programs is often protected using well-known and standardized protocol suites, such as the Transport Layer Security (TLS), which obscures the exact nature of the communication with encryption, forcing observers to infer it using metadata. The modern Internet is encryption heavy. As high as 90% of web browsing is protected by TLS. The newest version 1.3 is considered unbreakable by even the most well-resourced nation-state adversaries.

Although modern networks are packet-based, examining encrypted packets separately does not yield much information. On the other hand, combining all packets into a large pool and examining its properties does not grant much insight either. The useful middle ground is to leverage the connection-oriented nature of the communication, where applications establish sessions to exchange data. For example, the Hypertext Transfer Protocol Version 2 (HTTP/2) uses a request–response model, where one endpoint (client) initiates the connection and sends HTTP requests,

and the other endpoint (server) receives the HTTP requests and sends back HTTP responses to the client [1].[5] The next step in evolution, the Hypertext Transfer Protocol Version 3 (HTTP/3), now requires TLS and contains several anti-profiling techniques which seek to prevent application fingerprinting and metadata extraction [2, 101]. The adoption of this protocol is likely to hinder traditional approaches to traffic profiling and metadata collection, even for nation-state adversaries; an apparent design goal for HTTP/3.

Using encryption does not render network traffic completely unusable from the IDS standpoint. Network flows contain information that cannot be encrypted. In addition, the flows can be analyzed using features created by observing how and when the packets are transmitted [59]. Using specialized software, such as Suricata[6], it is possible to correlate individual packets and construct network streams where the packets are likely to be a part of one connection. These can be presented as time-series, where packet properties, such as the size, are combined with temporal properties, such as the arrival time. The features can then be used as a basis for statistical analysis and machine learning solutions, including neural networks. Malware does not usually contain sophisticated algorithms for generating traffic patterns that successfully evade advanced detection.[7] In addition, their functionality almost inescapably requires deviation from expected traffic patterns. These deviations may occur, for example, when the malware is instructed to exfiltrate data. By exploiting these shortcomings a network IDS can detect potentially malicious deviations from the norm. As the exact nature of the deviation cannot be ascertained by looking at the metadata, the process is called *anomaly detection*.

2.2.4 *Network traffic as time-series*

Network traffic is a man-made phenomenon, meaning we can take as close a look as we want to the processes, in both computing and statistical sense, that create it. We also know the rationale behind the design choices for each protocol, as well as the expected behavior under normal and error-induced conditions. In addition, malware analysis provides insight on how C&C channels are typically constructed. Using this knowledge is crucial when designing real-world security solutions.

In a statistical sense, the time-series arising from network traffic patterns are neither stationary nor linear (see e.g. [72] for formal definitions). The state of virtually any application is dependent on user input and influenced by factors such as other running programs, time of day, input data, or even pure randomness [32]. A network connection can remain relatively unused until the user performs an action, causing massive deviations from previously observed statistical properties (non-stationarity). As programs receive inputs from other sources than the network, only extremely limited predictions about the future behavior can be made using the data that the program has received (non-linearity). This behavior is completely expected and normal, yet it massively complicates or even prohibits the use of many traditional methods for time-series analysis.

[5] The internal workings of the protocol are more involved, as one connection may contain several bidirectional streams obscured by TLS-based encryption.

[6] https://suricata-ids.org/

[7] These would increase the size of the malware and threat of being detected by various endpoint protection solutions, such as antivirus applications.

Based on what we know about the networking protocols and the applications that use them, we can predict that there exists certain correlations and causations within a time-series, even though these events are not characteristic to the whole time-series, or correlate with other similar events within a series. A request usually warrants a swift response, even if it is not connected to other request-response pairs. If this response is unsolicited, missing, delayed, or unusual in size, it may signal an anomaly.

Just as using the assumption of stationarity with a non-stationary time-series leads to mixing of unrelated events, a fully connected neural network learns correlations that are known to be impossible or irrelevant due to the nature of networking protocols, or the input data. To prevent this from happening, the functions that the ANN is likely to learn must be restricted to those that are plausible, by using e.g. a suitable causal receptive field [64], or some another style [52] of external limiting.

2.3 Model fooling attacks and medical images

Machine learning methods are increasingly used in a medical setting, where they perform various kinds of *computer assisted diagnosis (CAD)* tasks, initial assessments, early detections of diseases, or augment and aid the work of a diagnostician by providing smarter tools that can highlight possible problems or just speed up the work flow. As machine learning solutions become integral parts of healthcare systems at national scale, they can be classified as critical infrastructure along with the rest of the essential healthcare system.

2.3.1 *On cancer*

Cancer is a group of diseases characterized by abnormal cell growth that leads to various disorders [42]. Normally the cells forming a tissue function and replicate under various rules and safeguards which allow the tissue to perform its function [42]. However, external or spontaneous factors may alter cell's DNA. If these alterations are inherited when the cell divides, and the mutation breaks the cells capability to be regulated or regulate itself normally, the result may be a neoplasm (tumor). Generally, if the neoplasm exhibits characteristics know as the "hallmarks of cancer", it has the capability to alter surrounding tissue in formidable ways, and even spread to secondary locations (metastasize) [23, 24]. As expected, the originating tissue, location, and the specific mutations of the neoplasm in question heavily influence how the disease is first detected, how it progresses, and what treatments are available. The various forms of cancer have different incidence rates (CIR), and these rates may vary depending on age, sex, and other factors. The importance of originating tissue is reflected in the nomenclature, as various tumors are named based on that tissue. Cancers with high CIR are of special interest, as systematic approach in detection and treatment has a large potential effect on outcomes. For example, according to 2020 OECD report, the expected incidence of breast cancer among women is 29%, and it accounts for 17% of female cancer deaths [63].

Cancer, in its many forms, continues to be the second leading cause of mortality in the EU, accounting for 26% of all deaths [63]. As the COVID-19 pandemic will temporarily skew the percentages, it will also challenges the healthcare system to continue effectively diagnosing and treat-

ing cancer diseases while responding to the pandemic. Increasing throughput by using machine learning solutions may help the healthcare system to respond to massively increased workloads.

2.3.2 *Machine learning in cancer detection*

When suspecting that a tissue may contain neoplastic growth, one of the ways to determine its properties is to actually extract a piece of that tissue and look at it with a microscope [99]. Various histological techniques may be employed for making important cellular features visible [55]. One of the fundamental features for classification of tumors is cellular differentiation and anaplasia [42]. Malignant tumors tend to lose both morphological and functional similarity to the originating tissue, making cells visibly different from their healthy counterparts. These changes include changes in size and shape, abnormal looking cell division, changes in cell nucleus that cause excessive staining during histological analysis, and the lack of orientation between cells as expected of the tissue type in question. Figure 2 is a picture of a tissue samples exhibiting infiltrative ductal carcinoma, a form of breast cancer.

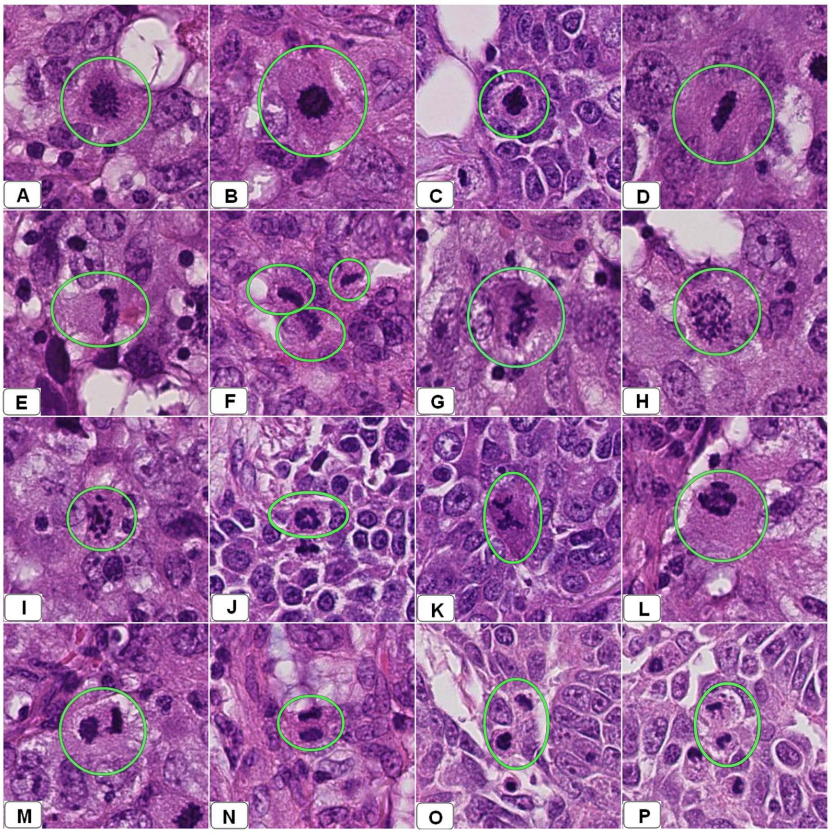


Figure 2: WSI showing several breast resections with infiltrative ductal carcinoma. Figure courtesy of Al-Janabi et al. [31], distributed under the terms of the Creative Commons Attribution License.

Digital photography allows pathologists to use computers for analyzing the histological samples. *Whole slide images (WSI)* refer to high-resolution digitized images of glass slides used in

light microscopy [67]. These images may contain multiple layers of differing zoom and focus levels.

WSI enable computers to process images, using various technologies such as traditional image manipulation, computer vision, and machine learning approaches. Machine learning models have been successfully used in cancer detection in the histological domain [38, 107]. The supervised models leverage hand annotated datasets to learn various metrics for tissue classification, such as mitosis, i.e. cell division count, and different kinds of abnormalities professionals have detected in the cells on the digitized full slide microscope image [38]. Although machine learning methods cannot fully replace the diagnostic decision making of a human professional, they can be used to perform computer assisted diagnosis [38]. Other tasks include finding regions of interest (ROI), by e.g. highlighting parts of images that are classified as abnormal [38].

2.3.3 *Model fooling*

Machine learning models are not perfect. They incorporate various biases and errors that stem from the entire spectrum of model creation. Usually just by selecting any machine learning technique, such as neural networks, we will introduce certain kinds of behaviors that will lead to unpredictable results in the problem domain. This is further enforced by the way the hyperparameters are tuned, the data is sampled, processed, and turned into features. These fragilities in machine learning models are exploitable. An attacker may use them to manipulate the machine learning solution into performing actions that lead to undesired results or loss of confidence in the solution itself.

Machine learning classifiers take an input, such as an image, and attempt to correctly sort it into one of the predefined classes. The aforementioned cat detector is a neural network classifier with two classes: cat and no-cat. We train it by procuring as many pictures of things both cat and no-cat as needed, until we deem it adequate. As expected, the model will in all likelihood fail to correctly classify certain cat-containing images, especially if they are markedly different from what was used in training. Cat orientation, lightning, framing, and other variables will, as expected, affect the accuracy of the model and predictions [4]. There are, however, other ways by which classification errors may happen.

Model fooling refers to the activity of taking a correctly classified sample, and altering it in a way which makes the model misclassify it with high confidence [69]. As altering may mean just swapping the sample image with another, we usually place additional constraints on how the sample may be altered. One of the most interesting choices for this restriction is to allow the manipulation of only one pixel of the sample, a so-called *one-pixel attack* [97, 98]. A human observer may fail to see any difference between the original and altered image. Vargas and Su suggest that the existence of one-pixel weaknesses are largely related to receptive fields [105]. Even though many problems are semi-discrete, minimizing a continuous function is far easier than a discrete one [44]. This may lead to unexpected behavior when an ANN is faced with samples containing values outside expected ranges of the legitimate input data. As it stands, the exact causes behind one-pixel attacks remains relatively unexplored.

Although this attack is usually demonstrated using pictures, it is just as applicable to many other problem domains. Misclassifying cats is usually harmless. In a more critical setting the cost of a misclassification can be significantly higher. For example, malicious altering of physical

objects, such as road signs, have the potential to disrupt self-driving cars that rely on machine learning [18]. Manipulating machine learning models in a medical setting is of interest to many adversaries. Attacks can range from insurance fraud, forging drug trial results, to other forms of relatively local misuse [19]. However when machine learning methods become commonplace, the healthcare system may ultimately be dependent on their correct operation. This exposes a new type of attack surface. At the time of writing there are no publicly known attacks against medical machine learning specifically. Unfortunately, when these misuses are revealed, they have usually been long ongoing.

3 RESEARCH CONTRIBUTION

This chapter presents the research contributions in chronological order, grouped by the thematic categories. First, papers concerning critical infrastructure are presented. Second, papers concerning machine learning and network intrusion detection are presented. Finally, the paper concerning medical images and model fooling is discussed. For each of the included articles, a short summary of the main elements is presented, along with the primary results. The chapter uses the term "method" broadly to describe the DSR approach, which may include several types of scientific inquiry. A short mention of the impact is also presented.

3.1 C1: Critical infrastructure and situational awareness

P1: Modelling and Real-time Analysis of Critical Infrastructure using Discrete Event Systems on Graphs

Aim. The objective of this study was to create a mathematical model for interdependencies and cascading faults in critical infrastructure. In addition, methods for quantitatively measuring the current and future state of CI after incidents were considered. The general design goal was to create a model that can include thousands of components, and still be fast enough for real-time applications.
Method. Critical infrastructure consists of systems and dependencies between them. After considering the nature and type of these dependencies, a graph theoretic approach was selected to model interdependencies [50, 80, 106]. For individual components, the approach taken was to leverage finite-state transducers for representing one CI component, such as an electrical transformer station. The states represent the operational status of the component, for example *OK*, *Fail*, and *Pre-Fail*. The transducers are connected to each other via a directed graph which represents the dependencies between separate components. When a component changes state, the symbol emitted by the respective transducer is broadcasted to every connected transducer, which changes their state accordingly. This may trigger further transitions, modeling a cascading failure. For assessing the impact of a particular event, each state in every transducer was equipped with a "badness" score. The criticality of each transducer was determined by a graph centrality measure that estimates how many components depend on that particular transducer, and how "central"

they are in terms of dependent components and their subsequent importance, as indicated by the centrality measure. Several metrics were defined to estimate the impact of an event: downstream weighted impact sum, a graph-centrality aware impact measure for events, and upstream risk, a measure that estimates how much risk is incurred by the failures in components that any particular component depends on. The performance of the model was evaluated with both simulated and real-world data from the open topographic database offered by the National Land Survey of Finland.
Results. The benchmark results indicate that the developed methods are capable of real-time performance at scales required for large infrastructures. The model was used in several research articles and technical reports, such as one commissioned by the Prime Minister's Office of Finland (VN TEAS) [30].

P2: Integrated Platform for Critical Infrastructure Analysis and Common Operating Picture Solutions

Aim. The objective of this study was to develop a framework for modeling, simulation, and analysis of critical infrastructure. The goal of the framework was the capability of assessing how various fault conditions and mitigation methods affect the severity of incidents via simulations. Specifically, human-in-the-loop decision making and SA considerations were included in the framework. This work was related to work commissioned by the Prime Minister's Office (VN TEAS), which included tasks to assess e.g. the effect of weatherproofing measures to storm resistance. The main goal of the framework was the suitability for this simulation task.
Method. The approach was to create a large-scale simulation model including 2G/3G/4G networks and electricity distribution networks. The simulation area was based on a real coastal area of Finland 50 km west of the capital Helsinki. The model included data from various sources, such as field measurements, open data, and expert interviews. The final model included an electricity distribution network, a multi-operator mobile communications network, building data from the Real estate, building, and spatial information database of the Digital and Population Data Services Agency, as well as 3D terrain models. Additional data was generously provided by Caruna Ltd. and other stakeholders.

The COP platform contained various visualization tools, as well as the modeling and analysis tools from **P1**. Using the analysis methods, the COP system could provide priority lists containing those infrastructure components that should be repaired first to maximize recovery. The simulator enters the list to a simulated repair queue. This models the human-in-the-loop behavior, where a human operator responds to faults using SA provided by the COP. The design is modular, and various parameters or alternative analysis methods can be benchmarked with little effort. Requirements were collected via expert interviews, consisting of personnel from different stakeholders, such as several utility operators, mobile network operators and various emergency service providers.
Results. The overall structure of the framework is presented in Figure 2 of **P2**. Three scenarios were run using the simulation and COP tools, one describing the area as it existed in 2016, and the second using predictions on how the area would be weatherproofed in 2030. The third scenario was a hybrid scenario consisting both the storm and a targeted cyberattack against remote

controllable medium voltage grid entities. The work was used as a part of the aforementioned VN TEAS report [30], where the scenario results are presented in detail.

P3: Nationwide critical infrastructure monitoring using a common operating picture framework

Aim. The objective of this study was to present both a theoretical foundation and practical solutions for creating a common operating picture system for monitoring large-scale infrastructures. The study consisted, in part, of assessing our prior work in larger context, as well as present a way to measure the SA using tests. The article was written at the end of a larger research project, TEKES Digital Security of Critical Infrastructures (Disci).
Summary of contents. The article describes the Situational Awareness of Critical Infrastructure and Networks (SACIN) framework, developed during the Disci project. The Joint Directors of Laboratories (JDL) data fusion model was used as a basis structure for the system [95]. The article details the theoretical framework, data collection and fusion, analysis methods, software architecture, and user interface design choices. The requirements for the system were based on expert interviews and other work conducted earlier in the research project [34, 51, 84, 85, 103]. The article details how the prior work can be structured using the JDL model, and developed using a situational awareness -oriented design process [15]. As the ultimate goal of a COP system is to provide SA, user tests are necessary in evaluating if there is an actual SA gained by using the system. The testing was conducted in two iterations, the first being [84], and the second one described here.
Method. The article details a set of visualization methods, including interactive and non-interactive variants. The following procedure was used to test if an inexperienced user could be familiarized with the system with little or no prior knowledge. A set of situational awareness measures were collected by having subjects ($N = 13$) complete trials. The participants were male graduate students attending a General Staff Officer course at the National Defence University (FIN). The test consisted of two 20-minute scenarios, one with an interactive interface, and one with non-active interface. The collected metrics, Situation Awareness Rating Technique (SART) [100], Situation Awareness Global Assessment Technique (SAGAT) [13], and System Usability Scale (SUS) [3] were compared. A detailed account of the statistical tests and results can be found in **P3**.
Results. The test results for SA differences between the two interface variants were mixed. Overall, the results support the conclusion that the system is able to increase operator SA. The article concludes that the JDL model is applicable to this problem domain. As the artifacts were developed using a situational awareness -oriented design process, the article concludes that the process can be used to identify SA requirements and translate them into designs that provide SA. Mica Endsley included the article in her meta-analysis on objective and subjective situation awareness [16].

P4: Blue Team Communication and Reporting for Enhancing Situational Awareness from White Team Perspective in Cyber Security Exercises

Aim. The objective of this study was to observe communication patterns during live cybersecurity exercises. Live cybersecurity exercises are dynamic in nature, requiring the exercise control (often known as the white team, WT) to have high levels of SA. The teams that practice defending cyber environments (blue team, BT) react to injects, i.e. pre-prepared events in the cyber range. When

observing an inject, e.g. malicious access to a system, BTs have to coordinate their response with each other via in-game communication tools, such as e-mail. WT needs to know how BTs respond and communicate for steering and pacing the exercise to fulfill the desired learning goals. In addition, after-action analysis of communication patterns may reveal critical flaws in real-life procedures or responses, as BTs are generally tasked to use them in exercises as well.
Data. Cybersecurity exercises are an important way to train the operators of various critical infrastructure fields to respond complex cyber attacks. Finland's National Cyber Security Exercise (kansallinen kyberturvallisuusharjoitus, KYHA) is an annual live training exercise, held since 2013. In 2017, the 4-day exercise was conducted by using Realistic Global Cyber Environment (RGCE), a cyber range developed by JAMK University of Applied Sciences Institute of Information Technology [33]. The exercise was attended by more than 100 individuals, forming 7 cooperating BTs [57]. The teams were given various common methods of communication. The study focused on e-mail communication, as it was preferred by the BTs. Due to confidentiality issues, the team names and e-mail counts ($N > 20000$, including various attacks) could not be reported in detail.
Method. The e-mail headers were extracted from in-game mail servers, and analyzed and visualized using Cytoscape[8]. Patterns were analyzed using graphs, where nodes are BTs and the edges show communication. Using timing information from e-mail headers, the communication patterns could be replayed, and correlated with various injects.
Results. After-action analysis of communication patterns revealed that for some teams the scenario was too light and did not provide adequate workload. Had WT been aware of this, the number or intensity of injects could have been adjusted. The patterns also revealed several omissions in communication made by training teams. Both findings suggest that communication pattern analysis is a beneficial tool for improving exercise outcomes. The paper also describes a custom reporting software tool that was created to facilitate communication between exercise control and training teams.

3.2 C2: Machine learning and network intrusion detection

P5: Anomaly-Based Network Intrusion Detection Using Wavelets and Adversarial Autoencoders

Aim. The objective of this study was to apply artificial intelligence and deep learning using ANNs to network traffic for detecting TLS-encrypted C&C channels used by APT-malware. The context for this work was a research project conducted for the Scientific Advisory Board for Defence (MATINE). The goal of the project was to research the applicability of artificial intelligence and deep learning using neural networks to IDS problem domain. This creates an obvious delimitation, as no other forms of detection were considered. Another delimitation was the choice of restricting the research to encrypted TLS traffic, as modern malware C&C channels and legitimate traffic in general use it. This delimitation was further warranted by the use of TLS in recent APT attacks.
Data. In 2018, the KYHA exercise was organized by The Ministry of Defence, The Security Committee, and JAMK University of Applied Sciences [58]. The exercise was conducted on The Realistic Global Cyber Environment (RGCE) cyber range [33]. We received permission to use the

[8]https://cytoscape.org

raw exercise data in this research. The KYHA18 dataset contains 729,998 TLS traffic flows, of which 665 flows are malicious. Malicious flows were generated by Meterpreter[9], Empire[10], and CobaltStrike[11] during the exercise. Benign flows contain both human and auto-generated web browsing traffic, such as authentication portal logins, automatic software updates, e-mail using TLS, and other common benign activity. The suitability of the dataset for IDS development is high due to human-generated traffic, although the scientific value is somewhat diminished by its confidentiality, as it is not approved for public release.

Method. The general approach was to transform TLS sessions into time-series, consisting of packet timing and size information. Haar wavelets [22] were used to convert the time-series into an image, that is then processed by a model based on neural networks. The wavelet approach was selected due to the non-stationary and non-linear nature of the input data. A neural network architecture based on adversarial autoencoders (AAE) was developed, and it was adapted to process decomposition results [52]. The AAE variant (TLS-AAE) created in the study was capable of clustering the results based on a similarity measure. The trained TLS-AAE outputs an anomaly score, quantifying how similar it thinks the input was with training data. The TLS-AAE was compared to traditional autoencoder. Following the RSD methodology, a working pipeline to process live network data was constructed to evaluate real-world suitability. The pipeline collects raw network data, and processes it into form suitable for TLS-AAE.

Results. Based on the findings, the combination of wavelet decomposition and adversarial autoencoders can detect anomalies, i.e. a selection of APT-tool traffic, with relatively good true positive rate (TPR 95%), although false positive rate (FPR 36%) remained characteristically elevated. The model performed better in comparison to traditional autoencoders. Furthermore, the requirement for real-time processing capability was satisfied. The methods detailed in the paper were tested using the Cobalt Strike BEACON payload that was used as a part of the SUNBURST attacks. The methods were successful in detecting the Cobalt Strike BEACON's command and control traffic.

P6: Network Anomaly Detection based on WaveNet

Aim. The objective of this study was to improve the results obtained in **P5**. The study focuses on the same type of TLS-encrypted command and control traffic. In addition, the goal was to include publicly available datasets to increase transparency, replication potential, and to allow comparison to other studies.

Data. The Intrusion Detection Evaluation Dataset (CIC-IDS2017) created by the Canadian Institute for Cybersecurity is one of the few modern publicly available labeled dataset containing full packet captures [90]. The dataset is fairly extensive, containing 1,425,742 flows, of which 1,107,695 were labeled benign, and 318,047 non-benign. However, the dataset did not contain virtually any TLS-based attacks. The benign 307,771 TLS flows could still be used. By generating additional malicious traffic using Empire and Cobalt Strike in the RGCE [33] environment, this dataset could be augmented to suit our evaluation purposes. The self-generated dataset included 15,124 benign flows (used to confirm that the environment is similar enough to CIC-IDS2017)

[9] https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/
[10] https://www.powershellempire.com/
[11] https://www.cobaltstrike.com

and 7,991 malicious flows. This dataset was created after the publication of paper **P5**, and was used only in **P6**. In addition, the KYHA18 dataset, described above, was used.
Method. Instead of transforming the time-series, we decided to use an architecture that could directly accommodate inputs of varying length, simplifying the process. Time-series were expanded to include packet direction, time difference to next received packet, time difference to next transmitted packet, and packet size. The final network included elements from WaveNet, Parallel WaveNet, and PixelCNN++ [64, 65, 86]. This customization was needed, as the original WaveNet did not have support for multiple features per time-step. As a bonus, this network architecture can provide insight on where in the packet timing sequence the anomaly occurs. The main hypothesis behind the selection was that the specialized temporal convolution in the WaveNet architecture could also work well with time-series data from networks. The network was tested using both KYHA18 and CIC-IDS2017 datasets. To allow comparison, the same datasets were used to benchmark TLS-AAE from **P5**.
Results. The overall results indicate that this approach outperforms the one presented in **P5**; TLS-AAE only scored 80% AUC whereas the new model got 91.61% AUC on the same KYHA18 dataset. Results of evaluation using CIC-IDS2017 dataset were markedly better, but this could be attributable to the lesser complexity of the data in the set. We utilized the data processing pipeline from the previous paper as the basis for a similar approach for assessing real-world performance with this solution. The methods were also successful in detecting the Cobalt Strike BEACON used in the SUNBURST attacks.

P7: Statistical Evaluation of Artificial Intelligence -Based Intrusion Detection System

Aim. The objective of this study was to provide a "sanity-checking" framework for neural network -based anomaly detection architectures. A neural network detecting anomalies must be sensitive to variations in certain patterns that are likely to vary between legitimate traffic and malicious C&C channels. In addition, the network should be resistant against fluctuations that are known not to correlate with malicious traffic. Failure to detect the changes in these patterns is taken as a sign that a neural network architecture cannot adequately measure useful characteristics of the traffic.
Method. The test consists of defining a statistical distribution that can output time-series with known statistical properties regarding correlations. The types of correlations are modeled after features we have discovered likely altering between malicious and non-malicious traffic: packet size, packet direction, and packet timing. The network from **P6** was trained with time-series samples drawn from the distribution. We then specify alteration to the original distribution, and draw "anomalous" samples from it. These "anomalous" samples are mixed with samples from the original to form ten sets where the percentage of "anomalies" is gradually increased from 10% to 100%. Each dataset is then evaluated using the network, and the mean anomaly score is recorded. We specify three types of alterations that test the sensitivity of the network.
Results. Based on the test results, the network was both able to detect altered time-series, as well as react in a stable fashion when the ratio between anomalous samples to normal samples is increased. The results suggest that the network architecture is sensitive to intended alterations, and the anomaly score behaves in a stable manner.

3.3 C3: Model fooling and medical images

P8: Model Fooling Attacks Against Medical Imaging: A Short Survey

Aim. The objective of this study was to conduct a short survey into model fooling attacks against medical machine learning systems. The aim was to map what types of attacks, if any, were successfully deployed against the models.
Method. This survey branched from the work the authors did for assessing the feasibility and novelty of article **P9**. A non-systematic literature review approach was selected due to the small pool of relevant publications.
Results. The survey revealed that the medical domain is relatively unexplored when it comes to attacks against machine learning classifiers. Only a few papers mentioning attack types in the medical context were found.

P9: One-pixel attack deceives automatic detection of breast cancer

Aim. The objective of this study was to create a practical one-pixel attack against a state-of-the-art machine learning classifier in a medical CAD setting. The goal of the attack was to construct a one-pixel perturbation which would flip a high-confidence classification of an input image to the other category, also with high confidence.
Data. We used a dataset from a machine learning competition, known as the Tumor Proliferation Assessment Challenge 2016 (TUPAC16) [54, 108]. The TUPAC16 dataset consists of 500 whole slide light microscopy images with known tumor proliferation scores, ground-truth labels, as well as region-of-interest data for 148 images. The chosen state-of-the-art classifier was IBM CODAIT Center for Open-source Data & AI Technologies' breast cancer mitosis detector [12]. The detector was chosen, as it achieved a high ranking with the TUPAC16 challenge data, and was released as open-source software.[12]
Method. The classifier is based on learning the morphological differences between healthy cells and possibly neoplastic variants. To find an image that can be suitably perturbed, a method based on differential evolution (DE), following the approach of Su et al. [96, 97], was used to alter candidate images, until the target perturbation was reached. The DE was used to search for two images. Starting with images containing abnormal mitosis, reach one where the classifier fails to detect this with high confidence. And conversely, starting with images containing normal mitosis, reach one where the classifier misclassifies it as abnormal with high confidence.
Results. Successful one-pixel attacks towards both directions were discovered (see paper for final images). Paper **P9** is, as far as the author can ascertain, the first publication presenting a one-pixel attack against a machine learning model used in medical CAD classification.

[12] https://developer.ibm.com/technologies/artificial-intelligence/models/max-breast-cancer-mitosis-detector/

4 DISCUSSION

The aim of this thesis was to consider critical infrastructure from several viewpoints, rather than focus on one narrow section. This thesis and the included articles address critical infrastructure from three different thematic categories: monitoring, defending, and exploitation.

Using design-science research methodology, the individual papers address specific problems primarily via developing an artifact in association with relevant stakeholders. In the framework of DSR, an instantiation of the artifact (e.g. a prototype, model, or method) demonstrates the feasibility of the product and the design process [26, 47]. This is contingent on rigorous evaluation of the design artifact, and a demonstration of its practical utility [70, 71]. This requirement is reflected in the central research questions, repeated below:

1. From one of the viewpoints, what salient problems does critical infrastructure have?
2. What are the real-life requirements for a suitable solution?
3. How do we acquire raw data from real systems?
4. How can we construct a functional prototype artifact?
5. Does the constructed prototype achieve the required real-life effect or performance?

4.1 C1: Critical infrastructure and situational awareness

One of the central questions surrounding critical infrastructure is how to maintain situational awareness over all the infrastructure sectors. Large-scale faults in CI are evolving situations, where active measures are taken to mitigate and restore the capability. For example, during a storm various field units are deployed to respond and prevent further damage. This presents three challenges: i) what must the operators who monitor the infrastructure be able to see for maintaining adequate SA for coordinating a response, ii) what components must be fixed first to restore as much capability as possible, or what must be protected to prevent massive cascading faults, and iii) how can this interaction be simulated and used as both a CI analysis tool, and as an event generator for training exercises, testing, and development, in order to facilitate the first two items.

Paper **P1** investigated the suitability of a graph-based mathematical model for cascading-fault analysis. In comparison to the methods presented in the literature, the model was purpose-built

for large-scale real-time models with thousands of dependencies and assets [66]. The combination of finite-state transducers modeling infrastructure components, directed graph interdependency modeling, and centrality measure -based analysis functions provided promising results when implemented as part of the simulator tool described in paper **P2**. Based on the data from real systems, received partly from various stakeholders, the relatively simple model formalism allowed constructing models based on a relatively small amount of information on the dependencies and internal operation of various components. This proved to be a requirement, as more detailed data was either proprietary, or unavailable. In addition, the model was deemed suitable for use in a technical report, where the goal was to assess how the telecommunication network and the electric grid would behave in different crises, and what impact foreseen future development would have [30]. Based on the findings, the report concluded that a weather-resilient electricity distribution network and battery backup systems are crucial in ensuring resiliency. Factors like the ability to reach emergency services via cell phones were considered. The approach was further refined by the author to include so-called entropy measures, that assign a time-dependent probability distribution to each transducer state [34]. This allows the model to account for the passage of time and its effects on analysis results, if no events concerning a particular component are received, for example due to damaged data networks.

Paper **P2** addresses the questions of CI simulation and analysis, while also considering how a common operating picture visualization system would improve SA, and how the simulator could be used to run scenarios for UX and user testing or training. Another goal for the paper was to provide a peer-reviewed basis for methods used in [30]. Overall, the solution provided reasonable results. Although direct comparison to a known gold-standard scenario was not possible, the proposed solution shows potential. More stakeholders should be included in the future to gather data spanning more CI sectors. Further validation could include comparison of simulator output to real data, although many stakeholders indicated that their systems do not currently collect suitable consolidated logs even from their own infrastructure components.

Paper **P3** presented a scenario-based SA test using volunteer personnel ($N = 13$), which returned mixed results. It is likely that the scenario was not challenging or detailed enough for distinguishing the arguably small difference between the two UX variants under examination. For this reason, the statistical results should be considered exploratory rather than confirmatory. The COP system requires status data from source CI components, and while connecting these components directly to the system would be the simplest approach, the risks of creating such a network outweigh the benefits. Furthermore, the information sharing must be transparent and fully controlled by the CI owner. An approach involving a middle-ware component called agent was deployed, as well as a platform for CI stakeholders to self-register their assets into the system. Based on expert interviews, the CI stakeholders were interested in sharing data between each other only if the integration method was flexible and customizable. This again reflects one of the design goals of **P1**, which can operate using extremely simplistic status data, and does not require CI operators to share sensitive details of their system.

Results in **P4** indicate that omissions in communication happen during incidents. Although the study considers a cybersecurity exercise, the participants were from companies that run CI. Deploying communication pattern analysis as a feature for a COP platform could improve after-action analysis also in real-world context.

The overall goal was to conduct research that had a clear practical use case in mind. In combination, the papers in C1 address the central research questions. Expert interviews and stakeholder data were both crucial elements in constructing the requirements, as well as in evaluating the feasibility and suitability of the resulting artifacts. The use of authentic data was paramount in achieving sufficient validity in the complex CI environment. The obtained results have already been applied to practical problems and decision making via [30]. Expert interviews and models based on real data improve the validity of the results. However, the lack of data access prevents running rigorous quantitative studies. Future work should focus on remedying this limitation, ideally by coordinating the effort with several CI stakeholders. Comparing the results with those in scientific literature remains difficult, as the approaches and goals are varied. Critical infrastructure has a major national element, and results obtained in other countries are not necessarily universally applicable. As it stands, traditional research designs and assessments of validity can rarely be deployed in research of this type.

4.2 C2: Machine learning and network intrusion detection

There are no free lunches in machine learning: prior assumptions are an inherent part of a working machine learning model [109, 110]. A central challenge in applying machine learning to intrusion detection is to select a suitable way to transform inputs and use a method that supports detection of features that are known to contain relevant components. This requires that the specifics of the problem domain must be carefully considered.

Most machine learning models and consequently also neural networks do not accept inputs of varying length. However, TLS connections can be extremely short, or very long-lived. Two different approaches were taken to address this. In paper **P5** Haar wavelet decomposition was used to transform packet timing and size information to an image. After a brief exploratory analysis on malware traffic, we concluded that the best approach was to aggregate the flows using a time window. Paper **P6** uses a neural network architecture capable of handling time-series of varying length. The latter approach benefits from being more fine-grained, although it still had a fixed maximum length.

The selection of training data was motivated by several factors, and evolved during the research project. In addition to considering the malware traffic, care must be taken to consider also the non-malicious traffic. If the non-malicious traffic is not varied or realistic enough, the validity of detection results remains low. As with all anomaly detection methods, data fusion and enrichment would drastically improve the performance in comparison of using just one detection method.

Selecting the type and architecture for the neural network has important consequences for time-series data. Papers **P5** and **P6** take different approach to this problem. The goal, however, remains the same: leverage a regularization method to prevent overfitting by exploiting what is known about the input data and the phenomenon in general.

The wavelet decomposition and adversarial autoencoder -based approach in paper **P5** used a continuous distribution for regularization. This approach yielded promising results, although the false positive rate remained high. Due to the complex nature of the network architecture, it proved to be challenging to track down the source for this, although we theorize that the dataset included

numerous non-malicious outliers. The use of wavelet decomposition to provide additional regularization proved to be novel, but capable of representing the nature of TLS flows. Overall, the architecture derived mixed results, despite the autoencoder paradigm's apparent applicability to the problem domain.

The enhanced time-series and multi-feature WaveNet approach in paper **P6** used causal convolution for regularization. This approach was markedly different from the one taken in **P5**. The decision to use an architecture that could directly accommodate inputs of varying length simplified the process, and removed one confounding factor.

The overall focus of the research was on assessing the suitability of neural networks for network anomaly detection. APT actors form the most insidious threat towards CI. In addition, advanced anomaly detection tools may require extra labor and expertise to be used effectively, limiting them mostly to protecting high-value targets such as CI. Focusing on the APT malware is therefore the priority choice. The benefits of encryption far outweighs the downsides of not being able to inspect traffic in transit.[13] As encryption is now basically a hard requirement for Internet-worthy remote management, IDS solutions must work under this constraint. As both methods require a significant amount of preprocessing, the task must be automated in a sufficiently scalable fashion. The data pipeline was constructed using open-source components, including Suricata IDS, TensorFlow framework, Kafka message bus, and Spark framework. This approach supports parallelization and scaling, and is capable of processing mirrored traffic in real time. In addition, the trained ANN models operate well within real-time constraints, and multiple instances of the same model can be deployed to increase parallelism. Considering production deployments is necessary even at the research stage, as it is an essential part of the viability assessment of the artifacts. Using well-known open-source frameworks increases compatibility and credibility.

IDS development rests on the availability of quality data. There are obvious practical and legal limitations preventing data collection from production networks.[14] While there are ways to automatically generate traffic, the interactions between users and cloud-enabled products are too complex to simulate. By using traffic from live cybersecurity exercises, we can capture full flows generated over four days by over a hundred people, as was the case in [58]. Independently creating traffic at this scale is unattainable to even large research groups, suggesting that cyber ranges like the RGCE [33] are instrumental in obtaining realistic raw data. As the attacks in live exercises are complex and directed towards systems that are in use, validity is further increased in comparison to test setups where a dummy target is created separately just to record the attack. Future research should focus on how to further take advantage of large-scale exercises, by e.g. increasing the variety of desktop programs and web services available to the users. Expanding the selection to contain entertainment, social media, and mobile applications will result in more varied traffic, as even the most demanding exercise contains low-intensity periods, and users are naturally drawn to these applications.

In general, there are numerous threats to validity in any research [89]. Although these are all too rarely mentioned in ANN studies, they play an extremely important role in this field as well.[15] Considering construct validity, the trend is to talk about "malicious" and "benign" data

[13] There are ways to allow middleboxes to decrypt and re-encrypt traffic, but these solutions are phased out as modern countermeasures against malicious interception prevent their function.

[14] Some of these limitations would also apply to IDSs in production.

[15] Failure to consider validity will result in what is elsewhere affectionately known as p-hacking [21].

points. These labels are assigned depending on if the data was generated by some malware or via harmless means such as web browsing. When talking about the actual measurements, the raw capture, or time-series constructed from TLS-encrypted flows, it is apparent that the measure does not fully reflect the construct of "maliciousness", as it is not directly tied to any effect of such sort. Conceivably, two identical packets or time-series could be labeled as both malicious and benign, depending on how the packet was created. Considering this inherent limitation, the time-series -based approach has a fair face-validity, in comparison to forms of measurements containing packet or payload -level details.[16] This assessment is based on what is know about APT malware and their C&C channels, as well as the the tactics, techniques and procedures of APT groups (see Chapter 2.2.1). Consequently, this transforms the meaning of "malicious" into "APT-like behavior" for the purposes of this research.

Broadly speaking, the goal of IDS ANN is to transform the input, and provide a score quantifying how "malicious" the input was. This transformation is learned primarily from the input data, but it is also influenced by the selected neural network architecture, and training procedures. Ideally, these are perfectly tailored to prevent the network from learning anything that is not directly related to the construct presented above. In practice, the network is selected based on what is considered to be the salient properties of the measurement, and general processing style that supports the construct, again based on what is known about APT actors. Although papers **P5** and **P6** take a different approach, both have properties that make them sensitive to types of variations presumed to be related to the construct above; the first having automatic clustering, and the second a causal convolution.

The statistical properties of the network data are complex. When combined with the fact that neural networks are hard to analyze, the test results, although promising, do not necessarily indicate that the networks are learning anything related to the construct above. This may happen, for example, if the input data still has non-construct -related differences between classes, and despite regularization the ANNs exploited it. To assess internal validity, a statistical testing method was devised (**P7**). The test consisted of defining a statistical distribution that can output a time-series with known statistical properties regarding correlations. The types of correlations are modeled after those we theorize are related to the construct above. A properly working IDS ANN should then be able to learn these correlations, while being insensitive to other forms of alterations. We then altered the input data by changing key correlations theorized to relate to the construct above, and observed the ANN output to see how sensitive the output is to these modifications. Although time constraints prevented running extensive test batteries and creating more sophisticated tests, we conclude that this procedure, in part, confirms that the network behaves as desired.

Thoroughly assessing external validity and generalizability would require additional datasets. In the studies, external validity is enhanced by using real or realistic environments and data: the generated traffic is what malware would generate on the systems that were its intended targets. This is also true for benign software. Considering ecological validity, recent APT campaigns, such as the one with SUNBURST, have deployed malware that would have been detected by the methods presented. This suggests that research utilizing open-source and commercial hacking tools is valuable against some of the TTPs used by APT groups. Continuous development would

[16] For a particular attack, these measures may have exceptional validity. In general, however, modern attacks have the payload obscured by encryption, and the attack is unrelated to low-level packet properties.

require continuous assessment of emerging APT tools. Automated procedures could be utilized to analyze packet captures as they are published on platforms like VirusTotal.[17]

A visualization tool was created to help development, and for assessing how the anomaly score reporting could be visualized in UI.[18] In addition, as a part of a report to MATINE, statements on the merits of project results were received from the Ministry of Defence, and the Finnish Defence Forces C5 Agency's Cyber Division. Both statements consider the results promising, with the FDFC5A recommending further research [37].

Perhaps surprisingly, AI practitioners sometimes face similar problems as psychologists do, when selecting research designs, choosing methods, and interpreting the results [91].[19] As usual in science, we must proceed with caution and carefully consider why we ended up with the result we did. As it stands, comparison to scientific literature remains challenging, as there are no commonly accepted testing and validation procedures, or benchmark datasets for modern APT threats. Moreover, using statistical methods for validity assessments is not common practice. In summary, however, all central research questions were explored in the context of this research theme, although to fully assess the prototype artifact, further research would be needed in collaboration with a CI network stakeholder.

4.3 C3: Model fooling and medical images

The third viewpoint of this thesis considers attacks against critical infrastructure. In particular, the goal was to find a novel exploit against emerging systems that are currently candidates for wide deployment in a relatively short time-frame. Studying them has the potential to impact the security of these deployments by informing the stakeholders about plausible ways to abuse this new infrastructure, and allowing them to create safeguards against malicious use. After consideration, machine learning in a medical setting was selected as target CI due to novelty and potential impact. This was followed by a brief survey into the current state of attacks against machine learning classifiers in medicine (**P8**), which revealed that this area is largely unexplored.

The results obtained in paper **P9** were not surprising, as similar results have been obtained in other problem domains [68, 105]. The practicality of the exploit is underscored by executing it via HTTP-API of a containerized application, a typical way modern Internet-facing web applications are deployed to cloud infrastructure. Consequently, this attack could be directed against an exposed API endpoint of a production application. Considering validity, a successful attack demonstrates that the hypothesis of the existence of such flaws was warranted. This further suggests that in general these flaws may prove to be common.

In comparison to previous themes, the produced artifact is considerably simpler. However, it demonstrates that a particular class of attack is feasible and straightforward to execute in the medical context as well. Although there was no direct collaboration with stakeholders during the studies, the system under attack was trained with real pathological data. Furthermore, the team responsible for the TUPAC16 challenge generously granted us access to the full dataset, including

[17] https://www.virustotal.com

[18] https://www.defmin.fi/files/4752/1245_MATINE-Tutkimusseminaari-JAMK.pdf

[19] Consequently, by learning from their successes and mistakes, AI as a field of science may yet avoid repeating the same statistical errors the field of psychology made in the beginning of the last century.

the ground truth. This collaboration allowed us to fully use the real data. As the literature suggests that vulnerabilities like these are likely to appear in any machine learning model of this type, the result does not indicate any particular flaw in this particular model or IBM CODAIT's general approach. Rather, it suggests that finding similar flaws in any such model is likely. Future research should focus on finding a suitable standardized test battery of perturbation types that could be run against any model. In line with previous themes, collaboration with commercial vendors of medical machine learning analysis software would yield results on the vulnerability of production systems, as well as solutions for hardening them against attacks of this nature. In conclusion, however, all central research questions were addressed in context of this research theme, albeit briefly.

4.4 Conclusion

This thesis considered three perspectives. Together, the studies illustrate how vast the field of CI is, and how diverse the problems are. However, there is a common theme to be seen here: the use of authentic data was paramount in achieving sufficient validity, and cooperation with stakeholders provided the needed insight on what the requirements and viable solutions really are.

Constructing a prototype artifact was successful in all three areas. Through that lens, the problems became tractable. The prototypes proved invaluable in demonstrating the solutions to the stakeholders, considering and evaluating the viability of the chosen approach, and just plainly seeing the solution really working. Although not without flaws, approach using the DSR methodology and prototype artifacts proved to be innovative, productive, and comprehensive. Continuous and strong national and international cooperation with our friends in industry, government, and academia is necessary for tackling the challenges of CI, as the whole is indeed greater than the sum of its parts. Just like critical infrastructure is.

YHTEENVETO (SUMMARY IN FINNISH)

Kriittinen infrastruktuuri muodostaa modernin yhteiskunnan kivijalan. Tähän infrastruktuuriin kohdistuu kuitenkin monia uhkia, joista tärkeimpiä ovat luonnonvoimat, ihmisten virheet ja erehdykset sekä tahallinen haitanteko, kuten kyberhyökkäykset.

Väitöskirjassa on kolme näkökulmaa. Ensimmäinen niistä käsittelee kriittisen infrastruktuurin matemaattista mallintamista, keskinäisriippuvuuksia ja tilannetietoisuutta. Osatutkimuksissa tarkasteltiin vikaantumisketjujen mallinnusta, sekä ennusteiden tuottamista mallintamisen avulla. Tutkimustuloksia hyödynnettiin Valtioneuvoston kanslialle tuotetussa selvitysraportissa, jossa arvioitiin erilasten vikatilanteiden, kuten myrskyjen ja kyberhyökkäysten vaikutusta sähkö- ja televerkon toimintaan. Lisäksi raportissa arvioitiin sähköverkon maakaapeloinnin vaikutusta verkon sietokykyyn ja kestävyyteen.

Toinen näkökulma käsittelee kyberhyökkäyksiä. Osatutkimuksissa hyökkäyksiä pyrittiin havainnoimaan tietoverkoista koneoppimisen ja neuroverkkojen avulla. Tutkimuksissa keskityttiin kehittyneiden nk. APT-toimijoiden hyökkäyksien paljastamiseen, sekä salattujen hattaohjelmakomentokanavien tunnistamiseen. Tällaisiin hyökkäyksiin lukeutuu muun muassa Solar Winds -tapauksessa käytetyt haittaohjelmat.

Väitöskirjan kolmannes näkökulma käsitelee kriittisen infrastruktuurin haavoittuvuuksia. Osatutkimukset käsittelevät terveydenhuollossa yhä yleistyviä automatisoituja diagnoosinaputyökaluja. Tutkimuksissa kehitettiin keino erheyttää neuroverkkoa, jota käytetään apuna syövän diagnosoinnissa. Kehitetyssä hyökkäyksessä solunäytteestä otettua kuvaa muokataan vain yhden kuvapisteen osalta siten, että neuroverkko tulkitsee terveen kudoksen virheellisesti sairaaksi tai toisin päin.

Osatutkimusten tuloksissa korostuu yhteistyön merkitys akateemisen maailman, viranomaistahojen ja infrastruktuurin omistavien yritysten ja yhteisöjen välillä. Yhteistyön kautta tutkimuksessa voidaan vastata relevantteihin kysymyksiin käyttäen reaalimaailman dataa ja asiantuntemusta.

REFERENCES

[1] M. Belshe, R. Peon, and M. Thomson,
"Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC Editor, RFC 7540, 2015.
[Online]. Available: http://www.rfc-editor.org/rfc/rfc7540.txt.

[2] M. Bishop, "Hypertext Transfer Protocol Version 3 (HTTP/3)",
Internet Engineering Task Force, Internet-Draft draft-ietf-quic-http-34,
2021, Work in Progress, 75 pp.
[Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34.

[3] J. Brooke, "SUS: A quick and dirty usability scale",
T. B. A. W. Jordan P. W. and I. L. McClelland, Eds., London: Taylor & Francis, 1996,
pp. 189–194.

[4] J. Buolamwini and T. Gebru, "Gender Shades: Intersectional Accuracy Disparities in
Commercial Gender Classification",
in *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*,
S. A. Friedler and C. Wilson, Eds., ser. Proceedings of Machine Learning Research,
vol. 81, New York, NY, USA: PMLR, 2018, pp. 77–91.

[5] G. Cassella and R. L. Berger, *Statistical Inference*.
Australia Pacific Grove, CA: Brooks/Cole Cengage Learing, 2001, 688 pp.,
ISBN: 0-534-24312-6.

[6] A.-L. Cauchy, "Methode generale pour la resolution des systemes d'equations
simultanees", *Comptes rendus de l'Académie des Sciences*, vol. 25, pp. 536–538, 1847.
[Online]. Available: https://ci.nii.ac.jp/naid/10026863174/en/.

[7] G. Cybenko, "Approximation by superpositions of a sigmoidal function",
Mathematics of Control, Signals, and Systems, vol. 2, no. 4, pp. 303–314, 1989.
DOI: 10.1007/bf02551274.

[8] Cybersecurity and Infrastructure Security Agency. "Advanced Persistent Threat
Compromise of Government Agencies, Critical Infrastructure, and Private Sector
Organizations". (2020),

[Online]. Available: https://web.archive.org/web/20210117173357/https://us-cert.cisa.gov/ncas/alerts/aa20-352a (visited on 2021-01-07).

[9] Cybersecurity and Infrastructure Security Agency. "JOINT STATEMENT BY THE FEDERAL BUREAU OF INVESTIGATION (FBI), THE CYBERSECURITY AND INFRASTRUCTURE SECURITY AGENCY (CISA), THE OFFICE OF THE DIRECTOR OF NATIONAL INTELLIGENCE (ODNI), AND THE NATIONAL SECURITY AGENCY (NSA)". (2021),
[Online]. Available: https://www.cisa.gov/news/2021/01/05/joint-statement-federal-bureau-investigation-fbi-cybersecurity-and-infrastructure (visited on 2021-01-07).

[10] J. I. Dingel and B. Neiman. "How Many Jobs Can be Done at Home?",
Becker Friedman Institute, University of Chicago. (2020),
[Online]. Available: https://bfi.uchicago.edu/wp-content/uploads/BFI_White-Paper_Dingel_Neiman_3.2020.pdf.

[11] A. Dresch, D. P. Lacerda, and J. A. V. Antunes Jr., *Design Science Research*.
Springer International Publishing, 2015. DOI: 10.1007/978-3-319-07374-3.

[12] M. Dusenberry and F. Hu. "Deep learning for breast cancer mitosis detection",
Center for Open-Source Data & AI Technologies (CODAIT). (2018),
[Online]. Available: https://github.com/CODAIT/deep-histopath/raw/master/docs/tupac16-paper/paper.pdf.

[13] M. R. Endsley, "Situation awareness global assessment technique (SAGAT)",
in *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference*,
vol. 3, IEEE, 1988, pp. 789–795. DOI: 10.1109/NAECON.1988.195097.

[14] M. R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems",
Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 37, no. 1, pp. 32–64, 1995. DOI: 10.1518/001872095779049543.

[15] M. R. Endsley, *Designing for Situation Awareness: An Approach to User-Centered Design*.
CRC Press, 2012.

[16] M. R. Endsley, "The Divergence of Objective and Subjective Situation Awareness: A Meta-Analysis",
Journal of Cognitive Engineering and Decision Making, vol. 14, no. 1, pp. 34–53, 2020.
DOI: 10.1177/1555343419874248.

[17] European Parliament and Council of the European Union, "Council Directive 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection",
Official Journal of the European Communities, 2008.

[18] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification",
in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2018.
DOI: 10.1109/cvpr.2018.00175.

[19] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning", *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019. DOI: 10.1126/science.aaw4399.

[20] FireEye Inc. "Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor". (2020), [Online]. Available: https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html (visited on 2021-01-07).

[21] J. K. Flake and E. I. Fried, "Measurement Schmeasurement: Questionable Measurement Practices and How to Avoid Them", *Advances in Methods and Practices in Psychological Science*, vol. 3, no. 4, pp. 456–465, 2020. DOI: 10.1177/2515245920952393.

[22] A. Haar, "Zur Theorie der orthogonalen Funktionensysteme", *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910. DOI: 10.1007/bf01456326.

[23] D. Hanahan and R. A. Weinberg, "The Hallmarks of Cancer", *Cell*, vol. 100, no. 1, pp. 57–70, 2000. DOI: 10.1016/s0092-8674(00)81683-9.

[24] D. Hanahan and R. A. Weinberg, "Hallmarks of Cancer: The Next Generation", *Cell*, vol. 144, no. 5, pp. 646–674, 2011. DOI: 10.1016/j.cell.2011.02.013.

[25] S. Haykin, *Neural Networks and Learning Machines*. Upper Saddle River, N.J: Pearson, 2009, ISBN: 0-13-129376-1.

[26] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design Science in Information Systems Research", *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004. DOI: 10.2307/25148625.

[27] C. F. Higham and D. J. Higham, "Deep learning: An introduction for applied mathematicians", *SIAM Review*, vol. 61, no. 3, pp. 860–891, 2019. DOI: 10.1137/18M1165748.

[28] S. Hollister. "Once again, someone tampered with an entire drinking water supply via the internet", The Verge. (2021), [Online]. Available: https://www.theverge.com/2021/4/5/22368476/kansas-man-tamper-water-supply-remote-ellsworth-wyatt-travnichek (visited on 2021-04-08).

[29] K. Hornik, "Approximation capabilities of multilayer feedforward networks", *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991. DOI: 10.1016/0893-6080(91)90009-t.

[30] S. Horsmanheimo, H. Kokkoniemi-Tarkkanen, P. Kuusela, L. Tuomimäki, S. Puuska, and J. Vankka, "Kriittisen infrastruktuurin tilannetietoisuus", *Valtioneuvoston kanslia*, 19/2017. [Online]. Available: http://julkaisut.valtioneuvosto.fi/handle/10024/160215.

[31] S. Al-Janabi, H.-J. van Slooten, M. Visser, T. Van Der Ploeg, P. J. Van Diest, and M. Jiwa, "Evaluation of mitotic activity index in breast cancer using whole slide digital images", *PloS one*, vol. 8, no. 12, 2013.

[32] D. Johnston, *Random number generators - principles and practices : a guide for engineers and programmers*. Berlin Boston: Walter de Gruyter GmbH, 2018, ISBN: 978-1-5015-1513-2.

[33] JYVSECTEC, "JYVSECTEC CYBER RANGE: RGCE and solutions",
JAMK University of Applied Sciences Institute of Information Technology, Tech. Rep., 2018. [Online]. Available:
https://jyvsectec.fi/wp-content/uploads/2018/10/JYVSECTEC-cyber-range.pdf.

[34] M. Klemetti, S. Puuska, and J. Vankka,
"Entropy as a metric in critical infrastructure situational awareness",
in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security, Defense, and Law Enforcement Applications XV*,
E. M. Carapezza, Ed., International Society for Optics and Photonics,
vol. 9825, SPIE, 2016, pp. 94–101. DOI: 10.1117/12.2219871.

[35] T. Kokkonen and S. Puuska, "Blue Team Communication and Reporting for Enhancing Situational Awareness from White Team Perspective in Cyber Security Exercises",
in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*,
O. Galinina, S. Andreev, S. Balandin, and Y. Koucheryavy, Eds.,
Cham: Springer International Publishing, 2018, pp. 277–288, ISBN: 978-3-030-01168-0.
DOI: 10.1007/978-3-030-01168-0_26.

[36] T. Kokkonen, S. Puuska, J. Alatalo, E. Heilimo, and A. Mäkelä,
"Network Anomaly Detection Based on WaveNet",
in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*,
O. Galinina, S. Andreev, S. Balandin, and Y. Koucheryavy, Eds.,
Cham: Springer International Publishing, 2019, pp. 424–433, ISBN: 978-3-030-30859-9.
DOI: 10.1007/978-3-030-30859-9_36.

[37] T. Kokkonen, M. Rantonen, S. Puuska, J. Alatalo, and E. Heilimo. "Tekoälyn käyttö poikkeamapohjaiseen tunkeutumisten havainnointiin verkkoliikenteestä",
MATINE. (2018), [Online]. Available:
https://www.defmin.fi/files/4498/2500M-0096_Tiivistelmaraportti_Kokkonen.pdf.

[38] D. Komura and S. Ishikawa, "Machine Learning Methods for Histopathological Image Analysis", *Computational and Structural Biotechnology Journal*, vol. 16, pp. 34–42, 2018.
DOI: 10.1016/j.csbj.2018.01.001.

[39] J. Korpihalkola, T. Sipola, S. Puuska, and T. Kokkonen, "One-pixel Attack Deceives Automatic Detection of Breast Cancer", *Computers & Security (under review)*, 2020.
eprint: arXiv:2012.00517.

[40] A. Kratsios, "The universal approximation property",
Annals of Mathematics and Artificial Intelligence, 2021.
DOI: 10.1007/s10472-020-09723-1.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks",
Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017. DOI: 10.1145/3065386.

[42] V. Kumar, A. Abbas, and J. Aster, *Robbins Basic Pathology*.
Philadelphia, Pennsylvania: Elsevier, 2018, ISBN: 978-0-323-35317-5.

[43] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, *Building high-level features using large scale unsupervised learning*, 2012. arXiv: 1112.6209.

[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[45] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539.

[46] A.-M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.

[47] T. Leinonen, T. Toikkanen, and K. Silfvast, "Software as Hypothesis: Research-Based Design Methodology", in *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*, ser. PDC '08, Bloomington, Indiana: Indiana University, 2008, pp. 61–70, ISBN: 9780981856100.

[48] T. G. Lewis, *Critical infrastructure protection in homeland security : defending a networked nation*. Hoboken, NJ: John Wiley & Sons Inc, 2020, ISBN: 9781119614531.

[49] S. Linnainmaa, "Taylor expansion of the accumulated rounding error", *BIT Numerical Mathematics*, vol. 16, 2 1976. DOI: 10.1007/bf01931367.

[50] E. Luiijf, A. Nieuwenhuijs, M. Klaver, M. van Eeten, and E. Cruz, "Empirical Findings on Critical Infrastructure Dependencies in Europe", in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2009, pp. 302–310. DOI: 10.1007/978-3-642-03552-4_28.

[51] L. Lääperi and J. Vankka, "Architecture for a system providing a common operating picture of critical infrastructure", in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, IEEE, 2015. DOI: 10.1109/THS.2015.7446228.

[52] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, *Adversarial autoencoders*, 2016. arXiv: 1511.05644.

[53] S. Mallat, "Understanding deep convolutional networks", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 2016. DOI: 10.1098/rsta.2015.0203.

[54] Medical Image Analysis Group Eindhoven (IMAG/e), *Tumor proliferation assessment challenge 2016*, http://tupac.tue-image.nl/node/3, 2016.

[55] A. L. Mescher, *Junqueira's Basic Histology: Text and Atlas*, Fifteenth Edition. McGraw-Hill Education, 2018, 576 pp., ISBN: 978-1-260-02617-7.

[56] Microsoft Corporation. ”Deep dive into the Solorigate second-stage activation: From SUNBURST to TEARDROP and Raindrop”. (2021), [Online]. Available: https://www.microsoft.com/security/blog/2021/01/20/deep-dive-into-the-solorigate-second-stage-activation-from-sunburst-to-teardrop-and-raindrop/ (visited on 2021-02-02).

[57] Ministry of Defence Finland. ” The authorities of the state administration are trained in cyber-skills in Jyväskylä - Valtionhallinnon viranomaiset harjoittelevat kyberosaamista Jyväskylässä 8.-11.5.2017, official bulletin 3th of May 2017”. (2017), [Online]. Available: https://valtioneuvosto.fi/-/valtionhallinnon-viranomaiset-harjoittelevat-kyberosaamista-jyvaskylassa-8-11-5-2017 (visited on 2021-04-05).

[58] Ministry of Defence Finland. ”The national cyber security exercises is organised in Jyväskylä - Kansallinen kyberturvallisuusharjoitus KYHA18 järjestetään Jyväskylässä, official bulletin 11th of May 2018”. (2018), [Online]. Available: http://valtioneuvosto.fi/-/kansallinen-kyberturvallisuusharjoitus-kyha18-jarjestetaan-jyvaskylassa (visited on 2021-04-03).

[59] A. Moore, D. Zuev, and M. Crogan, ”Discriminators for use in flow-based classification”, Queen Mary, University of London, Tech. Rep. RR-05-13, 2005.

[60] NASA, *Systems Engineering Handbook (NASA SP-2016-6105 Rev2)*. National Aeronautics and Space Administration (NASA), 2016.

[61] National Intelligence Council, Office of the Director of National Intelligence, *2021 Annual Threat Assessment of the U.S. Intelligence Community*. 2021. [Online]. Available: https://www.dni.gov/files/ODNI/documents/assessments/ATA-2021-Unclassified-Report.pdf.

[62] National Intelligence Council, Office of the Director of National Intelligence, *Global Trends 2040: A More Contested World*. 2021, ISBN: 978-1-929667-33-8.

[63] OECD and European Union, *Health at a Glance: Europe 2020: State of Health in the EU Cycle*. OECD Publishing, Paris, 2020, p. 230. DOI: 10.1787/82129230-en.

[64] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, *Wavenet: A generative model for raw audio*, 2016. arXiv: 1609.03499.

[65] A. van den Oord *et al.*, *Parallel wavenet: Fast high-fidelity speech synthesis*, 2017. arXiv: 1711.10433.

[66] M. Ouyang, ”Review on modeling and simulation of interdependent critical infrastructure systems”, *Reliability Engineering & System Safety*, vol. 121, no. 0, pp. 43–60, 2014, ISSN: 0951-8320. DOI: 10.1016/j.ress.2013.06.040.

[67] L. Pantanowitz, ”Digital images and the future of digital pathology”, *Journal of Pathology Informatics*, vol. 15, no. 1, 2010. DOI: 10.4103/2153-3539.68332.

[68] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, *Practical black-box attacks against machine learning*, 2017. arXiv: 1602.02697.

[69] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings", 2016. DOI: 10.1109/eurosp.2016.36.

[70] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research", *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007. DOI: 10.2753/MIS0742-1222240302.

[71] J. Pries-Heje and R. Baskerville, "The Design Theory Nexus", *MIS Quarterly*, vol. 32, no. 4, pp. 731–755, 2008. DOI: 10.2307/25148870.

[72] M. B. Priestley, *Non-linear and Non-stationary Time Series Analysis*. London San Diego: Academic Press, 1989, ISBN: 0-12-564910-X.

[73] T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection", Secure Networks Inc., Tech. Rep., 1998.

[74] S. Puuska, S. Horsmanheimo, H. Kokkoniemi-Tarkkanen, P. Kuusela, L. Tuomimäki, and J. Vankka, "Integrated platform for critical infrastructure analysis and common operating picture solutions", in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2017, pp. 1–6. DOI: 10.1109/THS.2017.8093737.

[75] S. Puuska, K. Kansanen, L. Rummukainen, and J. Vankka, "Modelling and real-time analysis of critical infrastructure using discrete event systems on graphs", in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2015, pp. 1–5. DOI: 10.1109/THS.2015.7225330.

[76] S. Puuska, T. Kokkonen, J. Alatalo, and E. Heilimo, "Anomaly-Based Network Intrusion Detection Using Wavelets and Adversarial Autoencoders", in *Innovative Security Solutions for Information Technology and Communications*, J.-L. Lanet and C. Toma, Eds., Cham: Springer International Publishing, 2019, pp. 234–246, ISBN: 978-3-030-12942-2. DOI: 10.1007/978-3-030-12942-2_18.

[77] S. Puuska, T. Kokkonen, P. Mutka, J. Alatalo, E. Heilimo, and A. Mäkelä, "Statistical Evaluation of Artificial Intelligence -Based Intrusion Detection System", in *Trends and Innovations in Information Systems and Technologies*, Á. Rocha, H. Adeli, L. P. Reis, S. Costanzo, I. Orovic, and F. Moreira, Eds., Cham: Springer International Publishing, 2020, pp. 464–470, ISBN: 978-3-030-45691-7. DOI: 10.1007/978-3-030-45691-7_43.

[78] S. Puuska, L. Rummukainen, J. Timonen, L. Lääperi, M. Klemetti, L. Oksama, and J. Vankka, "Nationwide critical infrastructure monitoring using a common operating picture framework", *International Journal of Critical Infrastructure Protection*, vol. 20, pp. 28–47, 2018, ISSN: 1874-5482. DOI: 10.1016/j.ijcip.2017.11.005.

[79] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He,
Zero: Memory optimizations toward training trillion parameter models, 2020.
arXiv: 1910.02054.

[80] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies",
IEEE Control Systems Magazine, vol. 21, no. 6, pp. 11–25, 2001. DOI: 10.1109/37.969131.

[81] H. Robbins and S. Monro, "A Stochastic Approximation Method",
The Annals of Mathematical Statistics, vol. 22, no. 3, pp. 400–407, 1951.
DOI: 10.1214/aoms/1177729586.

[82] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
DOI: 10.1037/h0042519.

[83] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
DOI: 10.1038/323533a0.

[84] L. Rummukainen, L. Oksama, J. Timonen, and J. Vankka,
"Visualizing common operating picture of critical infrastructure",
in *Next-Generation Analyst II*, B. D. Broome, D. L. Hall, and J. Llinas, Eds., SPIE, 2014.
DOI: 10.1117/12.2050231.

[85] L. Rummukainen, L. Oksama, J. Timonen, and J. Vankka,
"Situation awareness requirements for a critical infrastructure monitoring operator",
in *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*,
IEEE, 2015. DOI: 10.1109/ths.2015.7225326.

[86] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, *Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications*, 2017.
arXiv: 1701.05517.

[87] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers",
IBM Journal of Research and Development, vol. 3, no. 3, pp. 210–229, 1959.
DOI: 10.1147/rd.33.0210.

[88] J. Schmidhuber, "Deep learning in neural networks: An overview",
Neural Networks, vol. 61, pp. 85–117, 2015. DOI: 10.1016/j.neunet.2014.09.003.

[89] W. Shadish, T. Cook, and D. Campbell,
Experimental and Quasi-experimental Designs for Generalized Causal Inference.
Belmont, CA: Wadsworth Cengage Learning, 2002, ISBN: 0-395-61556-9.

[90] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*,
SCITEPRESS - Science and Technology Publications, 2018.
DOI: 10.5220/0006639801080116.

[91] J. P. Simmons, L. D. Nelson, and U. Simonsohn, "False-Positive Psychology", *Psychological Science*, vol. 22, no. 11, pp. 1359–1366, 2011. DOI: 10.1177/0956797611417632.

[92] H. Simon, *The sciences of the artificial*. Cambridge, Mass: MIT Press, 1996, ISBN: 9780262193740.

[93] T. Sipola, S. Puuska, and T. Kokkonen, "Model Fooling Attacks Against Medical Imaging: A Short Survey", *Information & Security: An International Journal*, vol. 46, no. 2, pp. 215–224, 2020. DOI: 10.11610/isij.4615.

[94] SolarWinds Inc. "SolarWinds Update on Security Vulnerability". (2020), [Online]. Available: https://orangematter.solarwinds.com/2020/12/17/solarwinds-update-on-security-vulnerability/ (visited on 2021-01-07).

[95] A. N. Steinberg, C. L. Bowman, and F. E. White, "Revisions to the JDL data fusion model", in *Sensor Fusion: Architectures, Algorithms, and Applications III*, B. V. Dasarathy, Ed., SPIE, 1999. DOI: 10.1117/12.341367.

[96] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces", *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997. DOI: 10.1023/A:1008202821328.

[97] J. Su, D. V. Vargas, and K. Sakurai, *Attacking convolutional neural network using differential evolution*, 2018. arXiv: 1804.07062.

[98] J. Su, D. V. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks", *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019. DOI: 10.1109/tevc.2019.2890858.

[99] P. Suonpää, R. Nevala, A. Beule, J. Schildt, G. S. Bova, and T. Mirtti, "Lähtökohdaltaan tuntematon syöpä", *Duodecim*, vol. 135, no. 9, pp. 838–846, 2019.

[100] R. M. Taylor, "Situational Awareness Rating Technique (SART): The development of a tool for aircrew systems design", in *AGARD Conference Proceedings No. 478, Situational Awareness in Aerospace Operations*, Copenhagen: Aerospace Medical Panel Symposium, 1990, ISBN: 92-835-0554-9. [Online]. Available: https://www.sto.nato.int/publications/AGARD/AGARD-CP-478/AGARD-CP-478.pdf.

[101] M. Thomson and S. Turner, "Using TLS to Secure QUIC", Internet Engineering Task Force, Internet-Draft draft-ietf-quic-tls-34, 2021, Work in Progress, 66 pp. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-quic-tls-34.

[102] J. Timonen, "A common operating room picture for dismounted operations and situation room environments", Ph.D. dissertation, Maanpuolustuskorkeakoulu, 2018. eprint: http://www.doria.fi/handle/10024/149449.

[103] J. Timonen, L. Laaperi, L. Rummukainen, S. Puuska, and J. Vankka, "Situational awareness and information collection from critical infrastructure", in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, IEEE, 2014. DOI: 10.1109/cycon.2014.6916401.

[104] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, *Deepfakes and beyond: A survey of face manipulation and fake detection*, 2020. arXiv: 2001.00179.

[105] D. V. Vargas and J. Su, *Understanding the one-pixel attack: Propagation maps and locality analysis*, 2019. arXiv: 1902.02947.

[106] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks", *Nature*, vol. 393, no. 6684, pp. 440–442, 1998. DOI: 10.1038/30918.

[107] M. Veta *et al.*, "Assessment of algorithms for mitosis detection in breast cancer histopathology images", *Medical Image Analysis*, vol. 20, no. 1, pp. 237–248, 2015. DOI: 10.1016/j.media.2014.11.010.

[108] M. Veta *et al.*, "Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge", *Medical Image Analysis*, vol. 54, pp. 111–121, 2019, ISSN: 1361-8415. DOI: 10.1016/j.media.2019.02.012.

[109] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997. DOI: 10.1109/4235.585893.

[110] D. H. Wolpert, "The Lack of A Priori Distinctions Between Learning Algorithms", *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, 1996. DOI: 10.1162/neco.1996.8.7.1341.

INCLUDED ARTICLES

ORIGINAL PAPERS

P1

MODELLING AND REAL-TIME ANALYSIS OF CRITICAL INFRASTRUCTURE USING DISCRETE EVENT SYSTEMS ON GRAPHS

by

S. Puuska, K. Kansanen, L. Rummukainen & J. Vankka 2015

2015 IEEE International Symposium on Technologies for Homeland Security (HST), 2015, pp. 1–5

https://doi.org/10.1109/THS.2015.7225330

Modelling and Real-time Analysis of Critical Infrastructure using Discrete Event Systems on Graphs

Samir Puuska*, Kasper Kansanen, Lauri Rummukainen, Jouko Vankka
Dept. of Military Technology
National Defence University
Helsinki, Finland
*email: samir.puuska@mil.fi

***Abstract*—Critical infrastructure (CI) systems form an interdependent network where failures in one system may quickly affect the state of other linked systems. Real-time modelling and analysis of CI systems gives valuable time-critical insight on the situational status during incidents and standard operation. Obtaining real-time quantitative measurements about the state of CI systems is necessary for situational awareness (SA) purposes. In this paper we present a general framework for real-time critical infrastructure modelling and analysis using discrete event systems (DES) on graphs. Our model augments standard graph-theoretic analysis with elements from automata theory to achieve model which captures interdependencies in CI. The framework was tested on various graphs with differing sizes and degree distributions. The resulting framework was implemented, and benchmarks indicate that it is suitable for real-time SA analysis.**

I. Introduction

Critical infrastructure forms a complex system where faults may quickly affect other systems and cause cascading failure chains. Real-time awareness of CI status and performance is a necessity for both everyday use, as well as for efficient incident response and disaster mitigation.

The research described in this paper is part of a larger project, called Digital Security of Critical Infrastructures (DiSCI). DiSCI project aims to find solutions for estimating and minimising threats facing the CI at national level. During the DiSCI project, the Situational Awareness of Critical Infrastructure and Networks (SACIN) framework was developed for evaluate CI monitoring concepts [1]. In the framework, the JDL data fusion model [2] was used for CI system integration. The research described in this paper covers the JDL levels 2 and 3 in the SACIN framework. Levels 2 and 3 are responsible for situation analysis and future risk estimation [1].

This paper continues with a summary of related work on section II. The model and its sub-parts are defined in section III. Section IV discusses the proposed analysis methods and risk estimation. The framework benchmarks are presented in section V. Finally, the conclusions and future work are discussed in section VI.

II. Related Work

Modelling and analysis of critical infrastructure is a notable field of contemporary research. It encompasses methods gathered from various different fields of science, including system dynamics, economic theory, and network theory, among others. [3]. Graphs have been previously used to model CI and its interdependencies. One notable graph-based approaches include the multi-graph system proposed by Svensen et al. [4], [5]. Graph centrality measures have also been studied in the context of critical infrastructure analysis [6].

Many of the current CI models do not especially address real-time requirements. In this paper, we present a model and and analysis framework where the real-time requirements have been taken into account. Our proposed model does not rely on extensive knowledge on the internal operation of CI systems or material flows and is therefore able to function with relatively modest amount of data.

III. Modelling Critical Infrastructure

Critical infrastructure consists of multiple interconnected systems. It is shaped by both the CI actors as well as the dependencies between them [7]. Modelling these systems as graphs captures these interdependencies in a way which enables quantitative analysis with existing and well established mathematical tools.

Critical infrastructure systems and actors are usually monitored closely for malfunctions and other deviations in their operational status. The monitoring is usually done by some automated system that is supervised by a human operator. When the operational state of some CI actor decreases we expect that it sends a notification to the dependent actors, and possibly to a centralised system. This fact, and the knowledge about dependency relations allows us to model CI as a set of communicating systems. Not all critical infrastructure systems are monitored. We can, however, indirectly observe their possible state by monitoring systems connected to them.

A. Critical Infrastructure Graph

We construct a graph that models the dependencies found between CI actors. The model assumes at least

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

partial knowledge of dependencies between CI systems.

Definition 1 (Critical infrastructure dependency graph): A directed graph $G = (V, E)$, where each vertex $v \in V$ represents CI actor and each edge[1] $e \in E$ a dependency relation between two actors. The graph is loop-free (no edges connecting vertex to itself), but assumed cyclic and disconnected.

In critical infrastructure system, events represent a change of capability in health of one CI actor. Events have indirect and sometimes delayed effect on all dependent systems. The delay is modelled as *critical time*, which determines how long a system can function without the actors which provide resources or services before its own capability or health drops. For example, some GSM base stations can operate three hours with battery power if power line goes down [8]. On the other hand, some systems such as diesel generators may start to offer a service (electricity) if they detect or sense that the main provider is malfunctioning.

Combining the concept of critical time and communication on state change allows modelling a critical infrastructure actors from a viewpoint that focuses on how systems react with respect to systems they rely on. The events emitted by an actor are propagated via paths in the dependency graph. Events may trigger delayed effects on systems, which model e.g. resource depletion.

B. Actor State Machines

Each vertex $v \in V$ is augmented with a state machine, which indicates the operational status of the corresponding CI actor. This models the health and capability states and transitions between them. The states should reflect clear reductions to ability to provide services outside.

We use *automata-based* approach for its intuitiveness, simplicity, and performance. Critical infrastructure is modelled as a set of interacting automata, where states change according to the output of other automata. More specifically, we base the model on Mealy machines, where the output alphabet represents outgoing message [9].

The *Actor State Machine* (ASM) is also augmented with support for delayed transitions, which model the critical time. This is achieved by assigning timed transitions to ordinary transitions, which are triggered if no other transition has taken place before the countdown.

Definition 2 (Actor state machine): The actor state machine is 8-tuple $ASM = (Q, \Sigma_i, \Sigma_o, T, O, D, S, q0)$, where

Q is a finite set of (capability) states;
Σ_i is a finite set of input events (alphabet);
Σ_o is a finite set of output events (alphabet);
$T : Q \times \Sigma_i \to Q$ is the transition function;
$O : Q \times \Sigma_i \to \Sigma_o$ is the output function;
$D : Q \times \Sigma_i \to Q \times \mathbb{R}^+$ is a delayed state transition function;
$S : Q \to [0, 1]$ is a status function.
$q0$ is the initial state.

The injective function S maps each state Q to some distinct real value between 0 and 1, which represents the relative "badness" of the state compared to other states. The badness is not formally defined and depends on the system an ASM attempts to model. The mapping is used to construct a state ordering which ensures that the model selects the worst-case scenario when conflicting input is entered. This can occur if, for example, two timed events trigger at the same time. This function is also used later in the analysis section. The value zero is the most undesirable state and one indicates full functionality.

A simple non-trivial ASM could contain three states $Q = \{\text{OK, PRE-FAIL, FAIL}\}$, shortened OK, PF and F. State OK means that the actor is capable of offering resources to the dependent systems. PRE-FAIL indicates that the actor is functioning correctly and able to deliver resources to its dependants, but one or more of its own dependencies are not functional, indicating possible upcoming failure in short timespan. FAIL means that the actor is incapable of producing necessary resources to sustain the actors which depend on it.

This simple automaton is illustrated in figure 1. The set of input and output events of this simple actor are $\Sigma = \{\text{OK, PRE-FAIL, FAIL}\}$, which allows chaining when Σ_i is Σ_o for some other automata. A simple automaton might also contain one timed transition, such as $(OK, PF) \to (F, 10)$, if the actor will transition from pre-fail to failed state after ten time units.

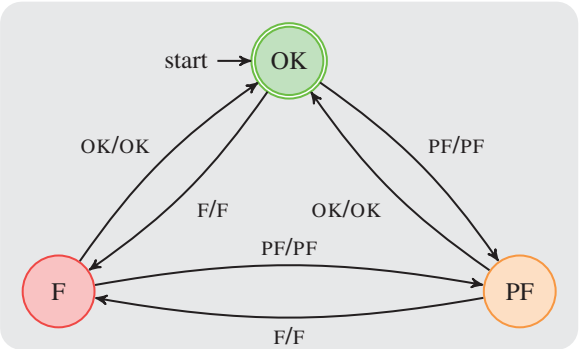


Fig. 1. Example actor state machine transitions and events. Edges indicate input / output, respectively.

C. Critical Infrastructure System

We model the CI system as a set of actor state machines that communicate with each other via events. These actors correspond to the nodes in critical infrastructure dependency graph, that may communicate via unidirectional first-in first-out *channels*, which correspond to the edges of the dependency graph, including direction.

[1]In this paper we use arc, edge, and dependency interchangeably.

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

Definition 3 (Critical infrastructure system): Critical infrastructure system is a 3-tuple $CIS = (G, \Sigma, q0)$ where

$G = (V, E)$ is a critical infrastructure dependency graph;
$V = \{\text{ASM}_1, \text{ASM}_2, \ldots, \text{ASM}_n\}$ is a finite set of vertices;
$E \subseteq V \times V$ is a finite set of directed edges (communication channels);
$\Sigma_{CIS} = \Sigma_{o_1} \cup \Sigma_{o_2} \cup \ldots \cup \Sigma_{o_n}$ is a finite set of events formed by union over all ASM input alphabets;

This system forms a *discrete event system*, where the state space is discrete and can be only altered via asynchronous discrete events. Events in this system may arise at any time. Any event $\sigma \in \Sigma_{CIS}$ may cause a state transition in some ASM. This state transition is propagated to dependant ACMs (representing CI actors) which, in turn, will cause them to change state.

IV. Analysis

The goal of the analysis is to assess the situation and health of the CI as a whole, and to quantify the impact of each discrete event on the infrastructure at its current state. Since the dependencies are modelled as a graph, we can use existing centrality measures to rank actors using their topological place in the infrastructure graph.

Traditional centrality measures typically consider the graphs topology in order to calculate nodes centrality. This is not enough in settings where nodes have changing attributes that affect the whole network. These effects may be permanent or temporary, and the timespan may vary. Static, pre-calculated centrality measures must be augmented with real-time data to respond changes in the network. Often the relative or absolute importance of a node cannot be calculated from a purely topological perspective.

Calculating centrality measures for all nodes is usually computationally expensive and can not be done in real time, except for very small graphs. The time complexity of many centrality measures is $O(|V||E|)$, number of vertices times number of edges [10]. By calculating the centrality (e.g betweenness centrality) for all nodes during initialisation phase, it's possible to use faster algorithms to scale the centrality in real-time, and avoiding costly centrality measure recalculations.

A. Updating the Actor State Machines

Updating actor state machines by traversing the graph can be done in $O(|V| + |E|)$ time complexity by expanding breadth-first traversal, as shown in algorithm 1. Because critical infrastructure dependency graphs are, in most cases, relatively sparse, and therefore $|E|$ is typically much smaller than $|V|^2$.

The update algorithm is run every time a new event arrives, or any ASM changes state due to a delayed transition. The source s is the node where some event σ causes a state change. The output event $e \in \Sigma_o$ is available for child nodes via `.symbol` attribute. The `sym[]` is a list containing all symbols (events) that parent nodes have produced. If a node has two direct parents, it will get two possibly different symbols. This conflict is resolved by `UpdateLevel`. The function tries every symbol in `sym[]` and sets the ASM to a state that has lowest output given by the status function S. The nodes store the old value of function S, as it is used later in analysis. The update algorithm considers only symbols that are available from parents at the time when one search depth level has been traversed. This means that back edges to upper levels are not used. Since most of the events do not propagate instantly through many layers of systems, this approach gives a large advantage over methods based on iterating the graph until the ASMs converge. It should be noted that events can still travel the graph in cycles, in some cases indefinitely, if they trigger timed transitions on their child nodes.

Algorithm 1: ASM update algorithm

```
input : A graph G and source vertex s
Result: Updated graph G
begin
    depth← 0;
    for each vertex v ∈ G do
        color[v] ← WHITE;
        d[v] ← ∞ ;
        sym[] ← NIL;
    color[s] ← GREY;
    d[s] ← 0 ;
    Q← ∅;
    P← ∅;
    ENQUEUE(Q, s);
    ENQUEUE(P, s);
    while Q ≠ ∅ do
        u ← DEQUEUE(Q);
        for each v ∈ child[u] do
            if (color[v] = WHITE) then
                color[v] ← GREY;
                d[v] ← d[u]+1;
                ENQUEUE(Q, v);
                if d[v] > depth then
                    UpdateLevel();
                    depth+1;
                v.sym[] ← u.symbol;
                ENQUEUE(P, v);
            else if (color[v] = GREY) then
                if d[u] < d[v] then
                    v.sym[] ← u.getSymbol;
        color[u] ← BLACK;
    UpdateLevel();
```

Function UpdateLevel

```
begin
    while P ≠ ∅ do
        u ← DEQUEUE(P);
        test each s ∈ u.sym[] and set ASM_u to state with
        lowest value given by function S;
```

B. Calculating Event Impact and Actor risk.

The impact of an event on the CI system depends on what systems it affects. We measure this effect by using

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

Downstream Weighted Impact Sum (DWIS). DWIS attempts to quantify the impact with pre-calculated centrality value and status value function assigned to each ASM.

DWIS is defined as

$$\mathrm{DWIS}(v) = \sum_{A_i \in \mathrm{T(v)}} \Delta S(A_i) \cdot C_i \qquad (1)$$

where v is the starting node, $T(v)$ is the set of all nodes reachable from v, $\Delta S(A_i)$ is the difference of ASM status value function before and after the state transition, and C_i is the (normalised) centrality of the node i. DWIS can be calculated in $O(|V| + |E|)$, since it requires only one pass through each affected node.

Each actor typically requires that all of the systems it relies on are operational at least with minimal capacity. If the systems suffer a reduction in capability or performance, it poses a threat for any system that requires those services or resources, because a low value of S might indicate that a total failure happens soon. We measure this threat with *Upstream Risk* (UR). UR attempts to quantify how much the performance of all the systems that are required by some specific actor is reduced.

UR is defined as

$$\mathrm{UR}(v) = \frac{S_1 + S_2 + \ldots + S_n}{N}, S_{1\ldots n} \in T^{\intercal}(v) \qquad (2)$$

Where S is the ASM status value function, $T^{\intercal}(v)$ is the set of all nodes reachable from v in the transpose (reversed edge directions) of CI graph, and N is the number of members in the set. The transpose can be calculated during initialisation phase.

V. Assessing the Performance Using Simulated Data

The proposed modelling and analysis tools were tested using different graphs generated with R using igraph package [11], [12]. The model and analysis framework was implemented using `Java` with `GraphStream` library [13].

Three types of random graphs were considered: totally random graphs generated with the Erdős-Rényi model, scale-free graphs generated with the Barabási-Albert model and small-world graphs generated with the Watts-Strogatz model. All graphs were generated in four different sizes, with node counts being 10, 100, 1000, and 10000. The Erdős-Rényi random graphs has a connection probability $p = 0.5$. The scale-free graphs were created with linear preferential attachment and zero appeal of one, giving the probability that a new vertex is connected to an old vertex i the probability $P(i) \sim k_i + 1$, where k_i is the out-degree of i. It should be noted that our graphs are the transpose of the original Barabási-Albert. The small-world graphs were first created as one-dimensional lattice graphs with every 2-degree neighbourhood fully connected. Then the edges were rewired with probability $p = 0.05$ and arbitrarily directed. The transposes of all graphs were also calculated. The small-world and scale-free graph types were chosen, because they are known to approximate real CI systems. [14]

TABLE I
Running time in milliseconds

node count	10	100	1000	10000
random	231	1168	111915	N/A
small-world	175	285	816	12655
scale-free	138	171	561	5238

Lastly, a graph depicting the power grid of Åland Island was constructed using the topographic database offered by the National Land Survey of Finland [15]. The graph has 812 nodes and 832 edges, with average degree distribution of 2.049. The map of the area is shown in Figure 3 and the resulting graph is displayed in Figure 2.

The benchmarks were run by randomly selecting a node from the graph. The node was fed a seed event that causes a state transition in all connected ASMs. The graph was updated and both DWIS and UR were calculated. This was repeated 1000 times for each graph and graph sizes. The results of generated graphs are illustrated on table I and edge counts on table II. Åland island benchmark took 1726 milliseconds. As expected, running time increases when edge counts grow larger. The software was unable to process totally random graph with 10000 nodes. The benchmarks were run on standard business laptop with Intel(R) Core(TM) i5-2520M CPU and 8GB of ram.

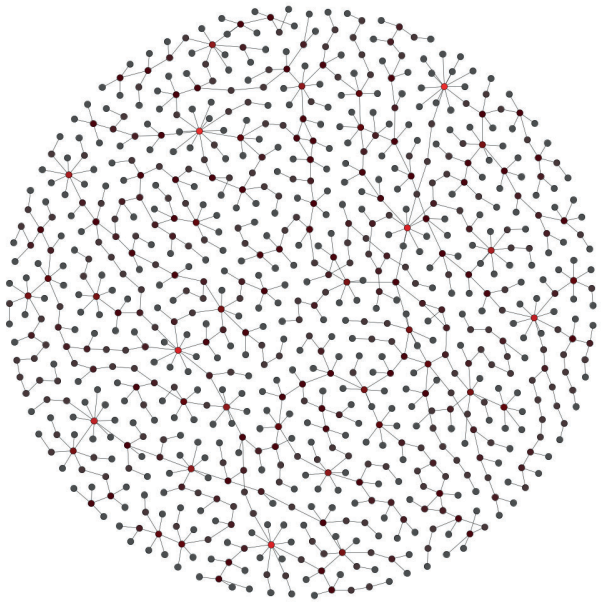


Fig. 2. Graph constructed from Åland island power grid

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

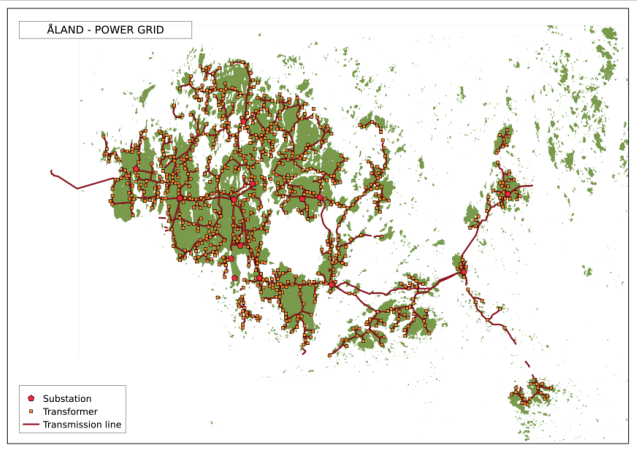


Fig. 3. Åland island power grid on map

TABLE II
NUMBER OF EDGES IN BENCHMARK GRAPHS

node count	10	100	1000	10000
random	50	4977	499378	N/A
small-world	20	200	2000	20000
scale-free	9	99	999	9999

Even though the interdependent nature of critical infrastructure is well established in literature, the empirical findings suggest that large scale failures that permeate many CI sectors are rare [16]. In real world single incidents do not usually cause failures that affect most of the network in a fashion these tests do. Therefore, in practice, the framework should perform faster when used with real-world data.

VI. CONCLUSIONS

In this paper we have presented a novel approach for CI modelling and analysis by combining graphs and customised finite state machines to produce a discrete event system. Analysis tools capable of real time operation were developed for assessing the possibility for implementing the framework as a part of CI monitoring solution. The framework was tested using several networks with varying size and topology. The results suggest that the framework is fast enough for real-time analysis.

A. Future Work

Our ongoing research focuses on both improving the event impact estimation and creating a more suitable centrality measures for critical infrastructure graphs. A centrality measure that uses domain-specific and tweakable parameters will deliver better results than purely topological approach. In the future the prospect of using approximations of centrality measures should be evaluated, as well as the possibility for on-the-fly topology changes.

The prospects of using real data and networks should be evaluated. For this purpose, a large-scale dataset(s) that span across multiple critical infrastructure sectors and covers both normal operation as well as faults of different type needs to be assembled.

REFERENCES

[1] J. Timonen, L. Lääperi, L. Rummukainen, S. Puuska, and J. Vankka, "Situational awareness and information collection from critical infrastructure," in *6th International Conference on Cyber Conflict*. NATO Cooperative Cyber Defence Centre of Excellence, 2014.

[2] N. A. Giacobe, "Application of the jdl data fusion process model for cyber security," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2010, pp. 77 100R–77 100R.

[3] M. Ouyang, "Review on modeling and simulation of interdependent critical infrastructure systems," *Reliability Engineering & System Safety*, vol. 121, no. 0, pp. 43 – 60, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832013002056

[4] N. K. Svendsen and S. D. Wolthusen, "Graph models of critical infrastructure interdependencies," in *Inter-Domain Management*. Springer, 2007, pp. 208–211.

[5] N. Svendsen and S. Wolthusen, "Multigraph dependency models for heterogeneous infrastructures," in *Critical Infrastructure Protection*, ser. IFIP International Federation for Information Processing, E. Goetz and S. Shenoi, Eds. Springer US, 2008, vol. 253, pp. 337–350. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-75462-8_24

[6] M. Barthelemy, "Betweenness centrality in large complex networks," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 163–168, 2004.

[7] S. Rinaldi, J. Peerenboom, and T. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *Control Systems, IEEE*, vol. 21, no. 6, pp. 11–25, Dec 2001.

[8] I. Horelli, "Tapaninpäivän 26.12.2011 myrskytuhot Lounais-Suomessa," Lounais-Suomen aluehallintovirasto, Tech. Rep., 02 2012. [Online]. Available: https://www.avi.fi/documents/10191/56990/Myrskyraportti+8.6.2012+LSAVI.pdf/5feb9ee3-426c-4806-99f7-220c2dd59955

[9] G. H. Mealy, "A method for synthesizing sequential circuits," *Bell System Technical Journal, The*, vol. 34, no. 5, pp. 1045–1079, Sept 1955.

[10] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[11] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org/

[12] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. Complex Systems, p. 1695, 2006. [Online]. Available: http://igraph.org

[13] GraphStream, *GraphStream*, RI2C research team, 2014. [Online]. Available: http://graphstream-project.org/

[14] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[15] Maanmittauslaitos, "Maastotietokanta," http://www.maanmittauslaitos.fi/en/digituotteet/topographic-database, 2014, [Online; accessed 10-2014].

[16] E. Luiijf, A. Nieuwenhuijs, M. Klaver, M. van Eeten, and E. Cruz, "Empirical findings on critical infrastructure dependencies in europe," in *Critical Information Infrastructure Security*, ser. Lecture Notes in Computer Science, R. Setola and S. Geretshuber, Eds. Springer Berlin Heidelberg, 2009, vol. 5508, pp. 302–310. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03552-4_28

978-1-4799-1737-2/15/$31.00 ©2015 IEEE

P2

INTEGRATED PLATFORM FOR CRITICAL INFRASTRUCTURE ANALYSIS AND COMMON OPERATING PICTURE SOLUTIONS

by

S. Puuska, S. Horsmanheimo, H. Kokkoniemi-Tarkkanen,
P. Kuusela, L. Tuomimäki & J. Vankka 2017

2017 IEEE International Symposium on Technologies for Homeland Security (HST), 2017, pp. 1–6

https://doi.org/10.1109/THS.2017.8093737

Integrated Platform for Critical Infrastructure Analysis and Common Operating Picture Solutions

Samir Puuska*, Seppo Horsmanheimo†, Heli Kokkoniemi-Tarkkanen†,
Pirkko Kuusela†, Lotta Tuomimäki†, and Jouko Vankka*

*Department of Military Technology
National Defence University, Helsinki, Finland
Email: samir.puuska@mil.fi

†VTT Technical Research Centre of Finland Ltd.
Email: seppo.horsmanheimo@vtt.fi

***Abstract*—In this paper, we present a software framework for modeling, simulation, and analysis of critical infrastructure (CI). Our concept fuses together a state-of-the-art telecommunications and electricity distribution system simulator (CI simulator), and a Common Operating Picture visualization system (COP system). The development process included expert interviews, which were conducted to define a comprehensive set of end-user requirements from different critical infrastructure stakeholders benefitting from a common situational picture. Using the obtained results, we enhanced the CI simulator to model more precisely interdependencies in communication and electricity distribution networks in normal and abnormal situations. In addition, the simulator was extended with near future prediction capabilities using the current situation and networks' operating conditions. The simulator also provides a real-time data stream to the COP system, whose core analysis and visualization functions were specified according to the end-user requirements collected from the interviews.**

I. Introduction

Modern society is becoming more and more digitalized and we are more dependent on communications and electricity in our daily life [1]. Electricity distribution and communications networks are two core critical infrastructure networks that form the base for the operation of modern society. Therefore, it is highly important to ensure that those networks stay operational in all situations, and that the recovery of the networks is as fast as possible. Co-operation and information sharing among the stakeholders is essential for offering reliable critical infrastructure services, fast recovery, and fault detection and mitigation. The high level situational awareness (SA) is mutually important for the operators of the critical networks, authorities, service providers, and end-users to understand what the provided information means and how to use it for proactive and reactive recovery operations in catastrophe situations.

A. Related Work

In our previous papers [2]–[5], we have presented our interdependency analysis process and a corresponding tool called NPT (Network Planning Tool) developed for redundancy analysis of commercial 2G/3G/4G networks in rural and urban areas. The tool was developed with real storm data from severe storms that have hit Southern and Northern parts of Finland in order to understand interdependencies between electricity distribution and mobile networks. In those papers, we were merely focusing on mobile network recovery techniques that could be used for improving resiliency of remote control of MV (medium voltage) grid entities.

In this paper, we extend the scope to assess impacts of a hybrid catastrophe to energy and communication networks as well as to citizens in present and future scenarios. The novelty is to connect a critical infrastructure network simulator with common operating picture system and be able to create near future forecasts.

II. Research Process

The main objectives were a) to create a large-scale simulation model including both networks at infrastructure and operational levels, b) to utilize field measurements for model construction, c) to provide a reliable and accurate snapshot of critical networks for different stakeholders about current and forthcoming situations, and d) to support decision-making of critical infrastructure stakeholders by presenting information in a more comprehensive ways [6].

Fig. 1 illustrates a scenario-based process that was used to achieve those goals. During the COP (Common Operating Picture) requirement phase, a large-scale and realistic threat scenario was specified at the target area. Based on recent public events, our catastrophe scenario included both natural (storm) and man-made (cyber-attack) threats. In our scenario, a severe storm damages electricity distribution and mobile networks entities. After the peak of the storm, a cyber-attack is attempted to cause damage in network entities and to disrupt communication and energy services.

The scenario was used to analyze technical impact, recovery, and information requirements for situational picture during and after the catastrophe when visibility and control over infrastructure components is reduced. In the context of the scenario, expert interviews spanning various organizations

978-1-5090-6356-7/17/$31.00 ©2017 IEEE

were conducted in order to gather information about interactions between different stakeholders, system requirements for situational picture as well as details about stakeholders' procedures, and technical methods used in large-scale disruptions (Section III). In addition, a review of future grid and communication technologies was conducted in order to model electricity distribution and communication networks also in the year 2030. Based on the interviews, technology reviews, and existing scientific knowledge on SA-design [7], a set of requirements for COP systems was defined.

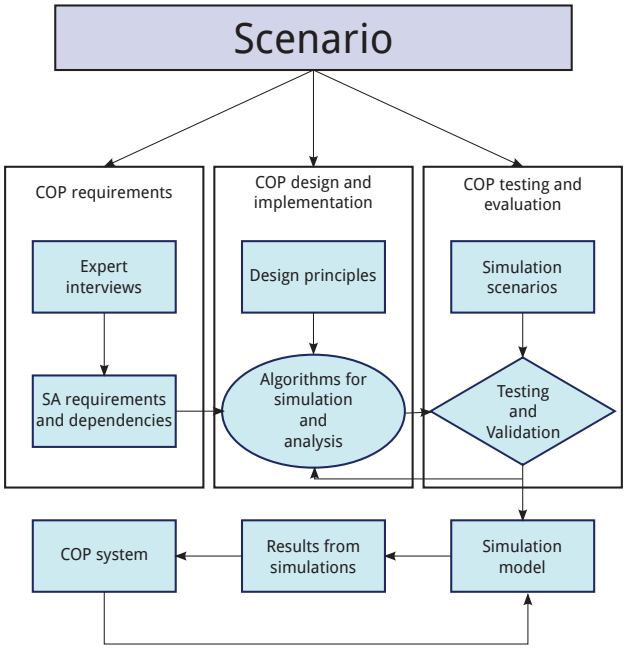


Fig. 1. The research process: Based on the set scenario, requirements are gathered and combined with existing design principles for situational awareness applications. Simulation scenarios are then validated and tested. This process creates the final simulation model.

During the COP design and implementation stage, a detailed infrastructure model of the target area was created (Section IV). The model includes components from electricity distribution network, mobile networks of multiple operators, and buildings including identified critical facilities such as hospitals and water treatment plants. Required simulation and analysis algorithms were developed to cover the present and future scenarios.

The COP testing and validation stage included correcting errors in CI models, and adjusting algorithms and parameters to enable more realistic simulations. The testing and validation was performed with two models, one reflecting the current (2016) infrastructure, and the other based on the prediction for year 2030 where e.g. underground power lines are more prevalent, mobile networks are more converged, and network automation allows more fine-grained error diagnostics and control.

The assessment of simulation results included a feedback loop between simulation model development, analysis of simulation results, and COP system development stages, as shown in Fig. 1. Real storm data from the target area was used to analyze the effects on the critical infrastructure in 2016 and 2030 scenarios. Examples of simulation results are presented in Section VI.

III. Expert Interviews

The conducted expert interviews covered different critical infrastructure stakeholders such as two major distribution system operators (DSOs), two mobile network operators (MNOs), and rescue service providers (rescue department, fire department and police). The interviewed persons were selected so that they were involved in the decision-making or field operations in the analyzed area. The interviews were conducted to assess the current state-of-the-art and the future insight of (C)OP systems in Finland. The interviewed persons were also asked what type of information they require or would like to obtain from other stakeholders as well as what data they are willing to share with their partners or customers. The interviews indicated that the most important requirements are the sharing of meaningful and up-to-date information, automated priority assignments, and accurate near future forecasts. In Finland, critical infrastructure is operated mainly by private or semi-private stakeholders, so a common data sharing platform is needed for efficient data sharing in catastrophe situations and for encouraging openness between CI actors. Based on the interviews, a set of requirements and use cases for both the CI simulator as well as the COP concept system were derived. In Table I, examples of general requirements for COP system are presented.

IV. Infrastructure Simulation

A detailed model of a coastal area of Finland was created. The area is located roughly 50 km west from Helsinki. The critical infrastructure models for years 2016 and 2030 were created and parametrized based on the interviews, technology reviews, open data sources, and field measurements. The critical infrastructure model included the structure of electricity distribution network with overhead and ground cablings, mobile communication networks of several operators, buildings with residential information, and 3D terrain model with clutter information. The model was designed to conform to the previously presented requirements, and it included the main CI components and their interdependencies. In addition to the dependency model, history data from a large-scale storm was used to model the fault and recovery events in the electricity distribution network. Fig. 2 illustrates how various data sources are used in the CI simulator and what outputs are available for the COP system and for post-processing. The state of the critical infrastructure is affected by fault events generated from scenario parameters. Based on the cascading faults, the simulator updates the networks' state and makes near future forecasts indicating what is going to happen in the next 2, 4 or 6 hours if nothing is done. This information is sent to COP system, which analyzes and visualizes the incoming data and returns priority lists indicating the most

978-1-5090-6356-7/17/$31.00 ©2017 IEEE

TABLE I
GENERAL REQUIREMENTS

No.	Requirement
1	Technical format is not important. It's essential, that the information is managed, analyzed, and aids decision making process.
2	Producing data for COP system is a joint effort. Every actor independently answers the production and validity of their respective fields.
3	Information must be preprocessed, analyzed, and understandable. It must have relevance for both sender and recipients.
4	Information should be presented clearly and visually.
5	Unnecessary technical details should be omitted. Information must be useful for experts from other fields.
6	COP system should be dynamic, and customisable to each infrastructure sector / user. Both high and low -level visualizations should be available.
7	System should have a map display, where multiple layers can be visualized separately and together. Clicking should provide extended details.
8	Data transfer should be automatic. Manual transfer causes unnecessary errors.
9	Terms and classifications should be consistent.
10	The use of a COP system should be included into the organization's processes such that it doesn't cause extra burden during incidents.
11	Organizations should be able to choose what kind of information they require and send.
12	COP system should be able to function using mobile networks and be portable.
13	COP system should enable information exchange between organizations and organizational levels.
14	COP system should be suitable for official and governmental use.
15	Information sharing should be possible between official and governmental organizations.
16	For maximal effectiveness, all infrastructure sectors should be represented in the COP system.
17	Information shared through COP system should contain a tag of original sender organization.
18	Not all information needs to be relevant to every user.
19	COP system should have capability for generating predictions for future development.
20	COP system should be capable of creating new information by combining incoming data.
21	COP system should dynamically prioritize systems and targets. These estimates should be manually adjustable.
22	Temporal dimension should be represented, containing information on general evolution of the situation (history and future).
23	Repair estimates should be available, and contain reason for fault.
24	System should be able to answer what, where, and why an incident has happened and what is the impact and repair estimate.

critical entities to be repaired in order to restrict the disaster area or to speed up the recovery.

V. ANALYSIS AND VISUALISATION

COP system is responsible for situation analysis and visualization. A set of analysis methods were implemented to be tested with the developed CI simulator and data sets. The model utilizes a graph-based approach for interdependency modelling and analysis. It estimates the impact of each infrastructure event, and suggests which devices should be repaired to gain largest increase in operational capability. The analysis framework utilized a set of methods previously published in [8]. The algorithms utilize directed graphs, where nodes represent CI components and edges dependency relations between them. Each node contains a weight factor that indicates its criticality. Criticality is based on several variables; how many components depend on a particular node, what are the priorities of the dependent components, and how redundant the topology is.

1) A general status display for each infrastructure sector, where a traffic-light-style ring is used to give a fast overlook on key subsections of each respective CI sector. Each ring is customizable for user's demands.
2) A list of events that the COP system has received is displayed, as well as their estimated impact on the whole system.
3) Another list is used to display what components should be prioritized (i.e. those that should be repaired first), as suggested by the analysis component.
4) A map display shows where the components are physically located, as well as their current operational status.

The map display also shows coverage of both mobile networks (per operator, or coverage type), as well as areas without electricity. It shows the priority of buildings, according to their usage type, such that e.g. hospitals and emergency service targets are of higher priority than ordinary residential buildings.

A strict demarcation between the CI simulator and analysis component was drawn to build a distributed CI/COP system such that the analysis component considers fault events generated by the simulator as if they were originated from actual critical infrastructure systems. The output of the simulator is sent as JSON (JavaScript Object Notation) based status messages to the prototype COP system. In the same manner, suggestions for recovery actions generated by the analysis algorithms are sent back to the CI simulator as JSON messages. This interaction models Human-in-the-loop behavior, where a decision made by a human operator assigns a repair priority to a particular target. The simulator can then mark components to be repaired based on prioritized suggestions provided by the COP system.

VI. SIMULATION RESULTS

Connecting the COP system to the CI simulator provides possibilities to analyze modified infrastructures, alternative technologies, new critical infrastructure services, and future scenarios. Furthermore, simulations allow to study the impact of e.g., weather-proof energy distribution, energy self-sufficient households, mobile base station batteries, remote control of energy substations, end-users' and services' telecommunication requirements, or co-operation of mobile operators in various kinds of fault situations. Such changes have impact on interdependent infrastructures as well as on

978-1-5090-6356-7/17/$31.00 ©2017 IEEE

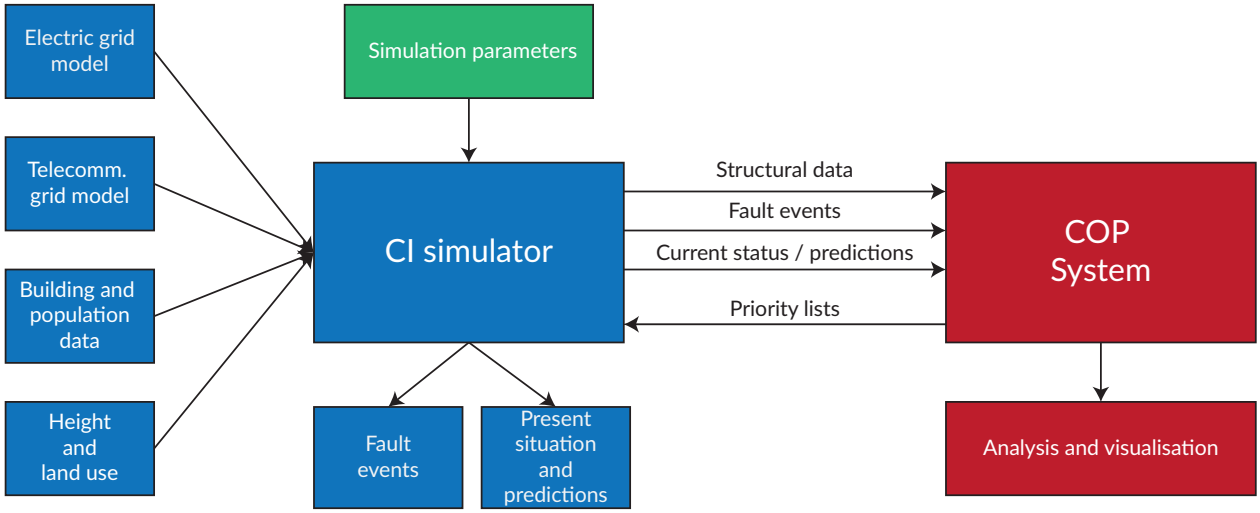


Fig. 2. Simulation structure and connections to COP system.

infrastructure users' ability to operate. As the use of remote operations and dependence on mobile services like real-time video transmission increase, telecommunications become more and more critical. With the proposed system, advanced features of COP can be integrated to infrastructure level events. For example, the CI simulator predicts in 2030 scenarios future mobile network coverage by taking into account the remaining battery lifetime (assuming no generators provided). The prediction of the forthcoming coverage can be shown at the CI or COP level, which offers proactive recovery capabilities to the current systems that experts are using for situational picture. Also, the infrastructure component repairs can be prioritized at the COP level and then fed back to the critical infrastructure management systems, thus providing higher level analysis covering several infrastructures contrary to infrastructures that prioritize their repairs independently of each other. This prioritization can be done in real-time during the disturbance.

Fig. 3 illustrates a summary of a storm scenario in 2016 (dashed) and 2030 (solid) electricity and telecommunication infrastructures as the storm progresses (x-axis). In 2030, the majority of overhead cables are replaced by weather-proof underground cables reducing the impact of the storm on electricity distribution. This can be observed from the percentage of operational secondary substations in 2030 (solid blue). Also, mobile base stations (red) benefit from secured electricity supply. The percentage of households with electricity (orange) in 2030 is high, because most people live in cities or towns where weather-proof underground cables are deployed. Although many base stations are down, emergency calls (purple) work throughout the storm. During a catastrophe situation, such pictures summarize temporal development over multiple infrastructure sectors.

The proposed CI/COP system supports decision-making during the disturbance by producing views at different levels

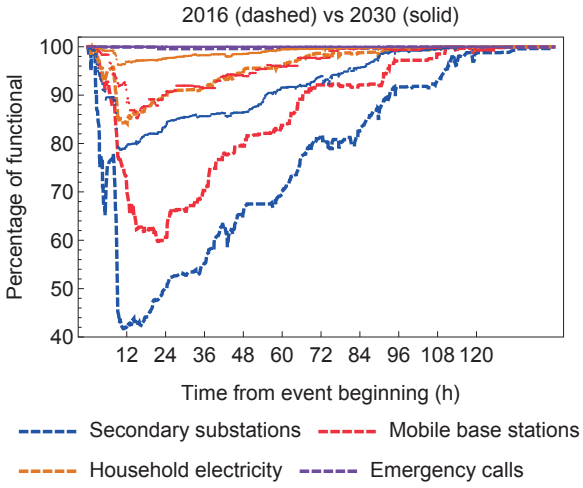


Fig. 3. Impact of storm on energy distribution, mobile communications and households as a function of time. Simulation scenarios of years 2016 and 2030.

and purposes. As an example of overviews, Fig. 4 and Fig. 5 are presented. Fig. 4 shows an overall picture of outages in electricity distribution network and their impacts on the area residents in year 2016 scenario. Buildings with and without electricity are illustrated using green and grey colors, respectively. The picture indicates that the impact of the storm is severe in rural areas. Only the urban regions, where underground cables are deployed, can maintain electricity distribution.

Fig. 5 shows a hybrid catastrophe situation in 2030 where a storm and cyber-attack have hit the electricity distribution grid and the cascading impacts can also be seen in telecommunication networks. The graph presents the number of non-operational base stations (brown), base stations without

978-1-5090-6356-7/17/$31.00 ©2017 IEEE

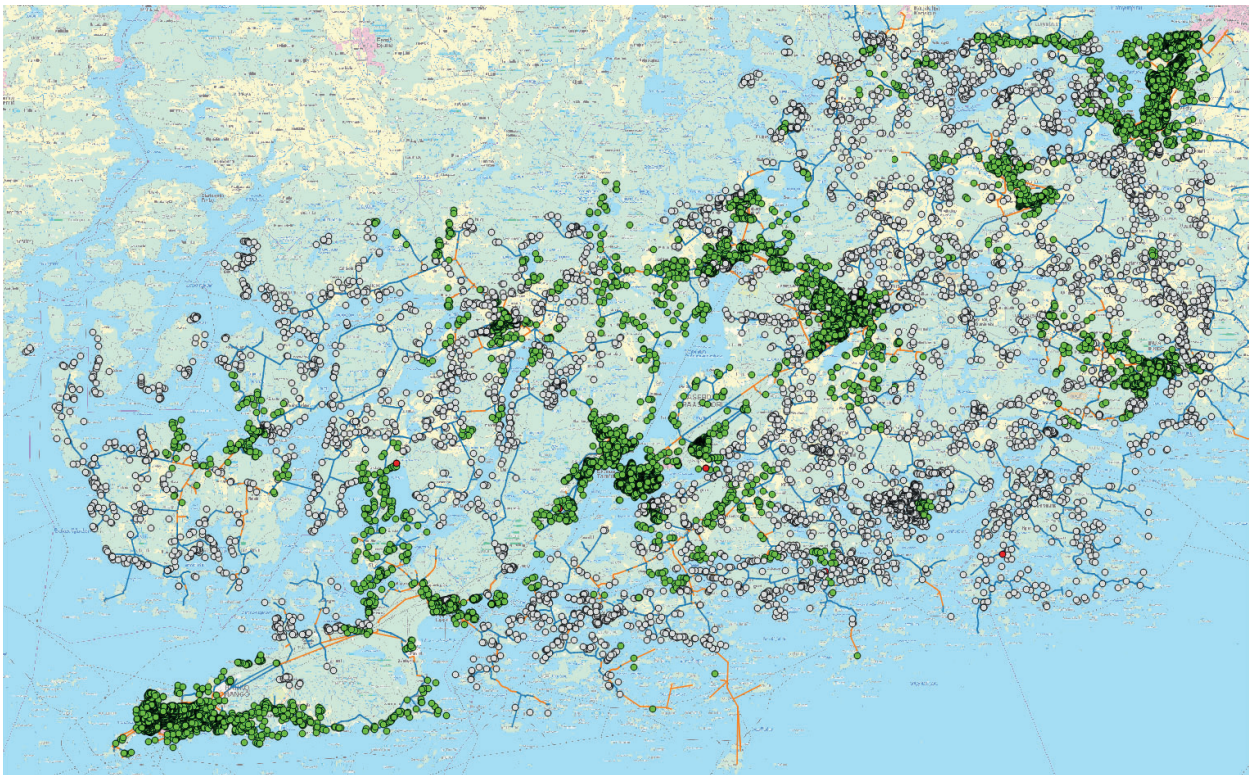


Fig. 4. Buildings with (green) and without (grey) electricity during a storm.

electricity (blue) and base stations on backup batteries (red). The profile shows two peaks, where the first one is caused by the storm and the latter one by the cyber-attack. The importance of base station batteries is seen clearly.

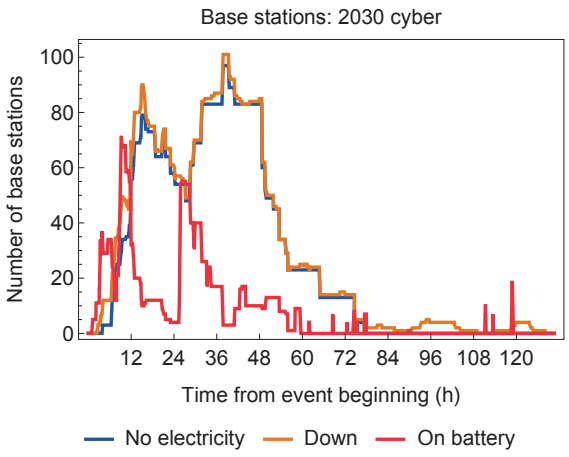


Fig. 5. Impact of storm and cyber-attack on mobile base stations in 2030 scenario.

Fig. 6 shows an example of more detailed view where situational picture is formed by combining data from several infrastructure networks and other sources. Buildings and electricity distribution network are shown on a map. Critical sites are presented with priority colors from green to red (the most critical) from the rescue services' viewpoint. This criticality changes dynamically according to the situation. The beige color shows areas without electricity, which means that generators are likely needed at the critical sites in these areas. The lilac color shows the areas where data communication is not possible. Either a dedicated terminal is needed or rescue operations must to be coordinated by using voice calls only. The figure shows also the buildings where residents without electricity and data cannot be notified by any online services

VII. Conclusions

The outcome of our work was an interconnected CI/COP system, which enabled us to assess interdependencies in critical infrastructure networks in realistic large-scale catastrophes. Additionally, the simulator was designed to be used as a fault and recovery event generator for COP systems, which allows the evaluation of different visualization techniques, analysis methods, and operating protocols in a controlled environment. Our emphasis was to provide meaningful information for decision-makers and to support proactive recovery actions with the aid of near future forecasts and dynamically changing prioritization lists. The implemented prototype helped us to assess existing and forthcoming interconnected critical infrastructure networks from decision-makers' viewpoint and to discover ways to speed up recovery or limit faulted areas with adequate resources. It is common that extensive repairs might

978-1-5090-6356-7/17/$31.00 ©2017 IEEE

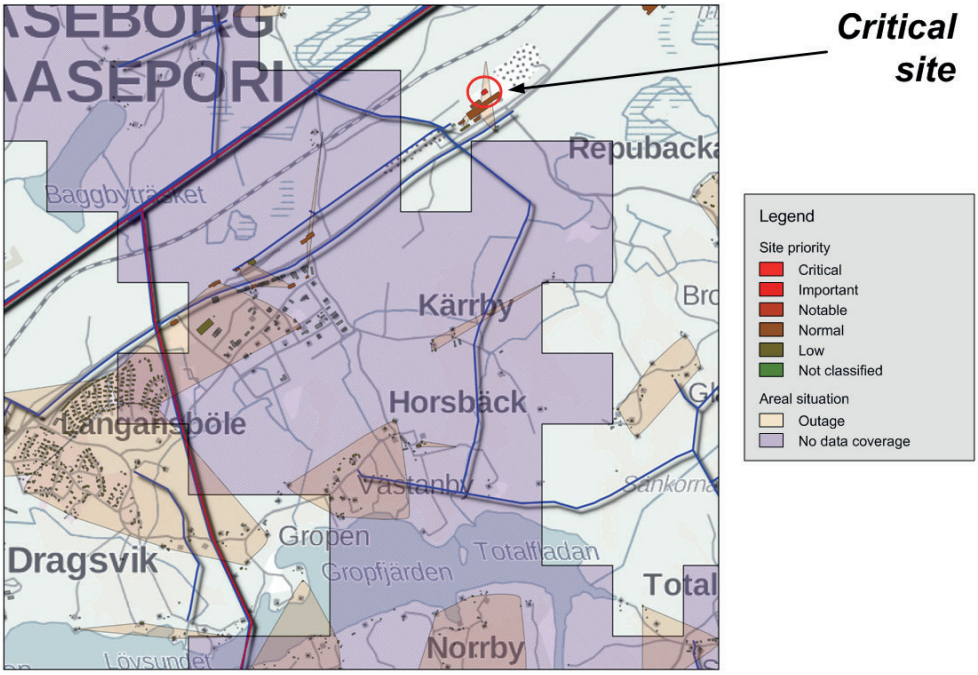


Fig. 6. Regions without electricity (beige) and data services (lilac) in a situation awareness view.

be required in rural areas before normal operational capability can be restored.

The presented framework forms a two-level interactive tool, which combines the operative network level with a high-level decision-making layer. This solution extends the traditional simulation and analysis approaches in two ways: firstly, by providing estimations of future outcomes, and secondly by combining network automation and human decision-making processes for a holistic assessment of resiliency of interconnected critical infrastructures. In addition, this framework allows rapid prototyping of COP solutions, such as analysis methods, user interface concepts, and execution of comprehensive user tests in a controlled and repeatable environment. It can also be used to simulate, how CI failures behave during large-scale incidents, such as storms or cyber-attacks, or any combination of these.

Acknowledgment

This work was done in "Kriittisen infrastruktuurin tilannetietoisuus" (Situational Awareness in Critical Infrastructure) project, which was funded by Finnish Prime Minister's office. We would like to thank experts from utilities, mobile operators, rescue and emergency service providers and, several ministries for their valuable contributions and comments to the research.

References

[1] T. G. Lewis, *Critical infrastructure protection in homeland security: defending a networked nation*. John Wiley & Sons, 2014.

[2] S. Horsmanheimo, N. Maskey, H. Kokkoniemi-Tarkkanen, P. Savolainen, and L. Tuomimaki, "A tool for assessing interdependency of mobile communication and electricity distribution networks," in *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE, 2013, pp. 582–587.

[3] S. Horsmanheimo, N. Maskey, L. Tuomimaäki, H. Kokkoniemi-Tarkkanen, and P. Savolainen, "Evaluation of interdependencies between mobile communication and electricity distribution networks in fault scenarios," in *Innovative Smart Grid Technologies-Asia (ISGT Asia), 2013 IEEE*. IEEE, 2013, pp. 1–6.

[4] S. Horsmanheimo, N. Maskey, and L. Tuomimäki, "Feasibility study of utilizing mobile communications for smart grid applications in urban area," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 2014, pp. 440–445.

[5] S. Horsmanheimo, N. Maskey, L. Tuomimäki, and K. Mäki, "Interoperability of electricity distribution and communication networks in large-scale outage situations," in *Telecommunications Energy Conference (INTELEC), 2015 IEEE International*. IEEE, 2015, pp. 1–6.

[6] S. Horsmanheimo, H. Kokkoniemi-Tarkkanen, P. Kuusela, L. Tuomimäki, S. Puuska, and J. Vankka, "Kriittisen infrastruktuurin tilannetietoisuus." Valtioneuvoston Kanslia, Tech. Rep. 19, 2017. [Online]. Available: http://vnk.fi/julkaisu?pubid=16803

[7] M. R. Endsley, "Situation awareness-oriented design," *The Oxford handbook of cognitive engineering. Oxford University Press, New York*, pp. 272–285, 2013.

[8] S. Puuska, K. Kansanen, L. Rummukainen, and J. Vankka, "Modelling and real-time analysis of critical infrastructure using discrete event systems on graphs," in *Technologies for Homeland Security (HST), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1–5.

978-1-5090-6356-7/17/$31.00 ©2017 IEEE

P3

NATIONWIDE CRITICAL INFRASTRUCTURE MONITORING USING A COMMON OPERATING PICTURE FRAMEWORK

by

S. Puuska, L. Rummukainen, J. Timonen, L. Lääperi,
M. Klemetti, L. Oksama & J.Vankka 2018

International Journal of Critical Infrastructure Protection, vol 20, pp. 28–47

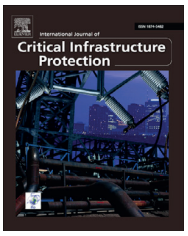
https://doi.org/10.1016/j.ijcip.2017.11.005

INTERNATIONAL JOURNAL OF CRITICAL INFRASTRUCTURE PROTECTION 20 (2018) 28–47

Available online at www.sciencedirect.com

ScienceDirect

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



Nationwide critical infrastructure monitoring using a common operating picture framework

Samir Puuska, Lauri Rummukainen, Jussi Timonen, Lauri Lääperi, Markus Klemetti, Lauri Oksama, Jouko Vankka*

Department of Military Technology, National Defence University, Helsinki PL 7, 00861, Finland

ARTICLE INFO

Article history:
Received 30 September 2015
Revised 14 August 2017
Accepted 15 November 2017
Available online 6 December 2017

Keywords:
Critical Infrastructure
Situational Awareness
Common Operating Picture
User Interface
Modeling and Analysis
Software Architecture

ABSTRACT

This paper describes the efforts involved in designing a common operating picture system for monitoring large-scale critical infrastructures. The design leverages the Joint Directors of Laboratories (JDL) data fusion model to enable the integration of different critical infrastructure systems with their dependency relations. The resulting Situational Awareness of Critical Infrastructure and Networks (SACIN) framework offers a platform that provides a common operating picture of a critical infrastructure. A generic data collection component customized to each source system generates events and facilitates JDL level 0 integration. An analysis component collects events and data to produce meaningful information about the current state and future impact estimates in accordance with JDL levels 1 to 3. A brokered architecture supports level 4 control by various components and a JDL level 5 user interface is offered via a web application. Interviews of infrastructure subject matter experts were conducted to obtain the situational awareness requirements. By applying key situational awareness oriented design principles to the situational awareness requirements, a user interface was created for organizing information based on operator situational awareness needs and supporting key cognitive mechanisms that transform data into high levels of situational awareness. Situational awareness measures were used to assess operator performance during critical infrastructure tasks – a "freeze-probe" recall approach (Situational Awareness Global Assessment Technique (SAGAT)), a post-trial subjective rating approach (Situational Awareness Rating Technique (SART)) and the System Usability Scale (SUS). The results indicate that the supply of attentional resources (SART supply) and overall SAGAT score best predict the performance levels of operators.

1. Introduction

Modern society relies on critical infrastructure systems that provide essential services. In order to effectively respond to failures and attacks on critical infrastructure at the national scale, it is vital to have situational awareness (SA) of all the infrastructure sectors.

The Situational Awareness of Critical Infrastructure and Networks (SACIN) framework described in this paper was developed for monitoring a diverse critical infrastructure environment. The SACIN framework incorporates data collection, fusion and integration steps that refine individual event streams into an operating picture. SACIN gathers information from various industrial systems via specific agent-

* Corresponding author.
E-mail address: samir.puuska@mil.fi (S. Puuska).

https://doi.org/10.1016/j.ijcip.2017.11.005

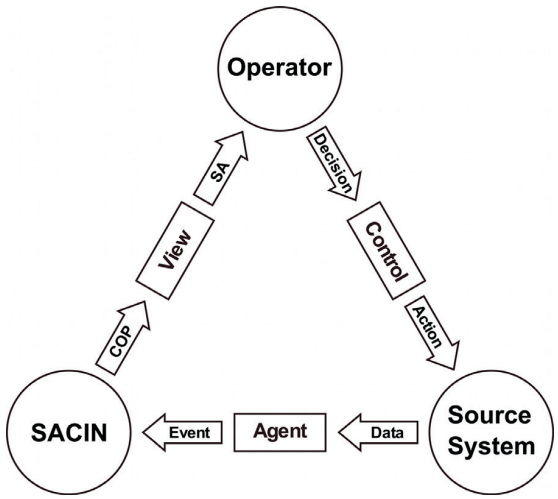


Fig. 1 – High-level view of the SACIN information loop.

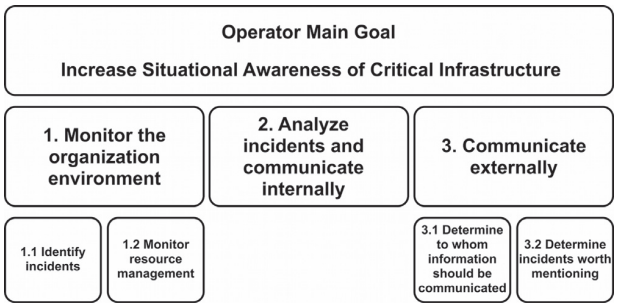


Fig. 2 – Goal hierarchy for a critical infrastructure monitoring operator [38].

based client-server interactions, processes and analyzes the gathered data, and displays the results in the form of a common operating picture (COP) [43].

Fig. 1 shows a high-level information loop corresponding to an operator controlling a critical infrastructure system such as a local electric grid. The source system is integrated with SACIN using a customizable middleware component called an agent. The agent processes messages and delivers essential information to SACIN. The SACIN analysis component uses events from multiple sources to create a model of the current state of the infrastructure. The operator is then presented with a view of the state of the critical infrastructure that supports comprehensive situational awareness.

This paper describes the efforts involved in designing the SACIN common operating picture system for large-scale critical infrastructure monitoring. It defines the system requirements and describes the main concepts and system architecture. The implementation is presented with a special focus on operator usability and performance evaluation. The Situational Awareness Global Assessment Technique (SAGAT), Situational Awareness Rating Technique (SART) and System Usability Scale (SUS) are applied and compared in order to evaluate operator performance of critical infrastructure monitoring tasks.

2. Related work

Critical infrastructure protection has become a major research topic in the past decade. As a result, a number of critical infrastructure modeling and simulation techniques have been developed. Ouyang [31] has conducted an extensive review of critical infrastructure modeling techniques. Examples of critical infrastructure modeling and simulation methods are presented in [5,6,10,19,28,29]. The research covers approaches that engage diverse perspectives such as ontological modeling, mathematical approaches, interdependencies and critical infrastructure services.

Despite the large body of research in critical infrastructure modeling and analysis, most of the approaches focus on small-scale systems with detailed views of the systems and are not designed for situational awareness purposes. Such a perspective is inadequate for providing a common operating picture of a large and complex critical infrastructure. The model presented in this paper attempts to address this issue by considering the ability to model a large number of systems. Since a common operating picture system has to operate in real time, the model must take into account the computational complexity of the update and analysis procedures, an aspect that is largely overlooked in the research literature.

Table 1 – Distinctive information blocks used in the final situational awareness requirements [38].

Situational awareness level	Incident	System or service
Level 1: Perception	Short description Time of occurrence Location	Location Purpose Contact information Resource requirements
Level 2: Comprehension	Magnitude Reason for incident Relation to others Whom to contact Reliability	Operational status Security status Priority and criticality System and service dependencies
Level 3: Projection	Duration (or estimation) Trend Effect	Sufficiency of critical resources

Table 2 – Situational awareness requirements for a critical infrastructure monitoring operator [38].

Always relevant	Ongoing incidents Current services and systems in the environment Affected services and systems
1.1 Identify incidents	Priority order of incidents Possible future incidents
1.2 Monitor resource management	Validity of incidents Service and system changes due to incidents
2. Analyze incidents and communicate internally	Contact information of relevant decision-makers Possible future incidents Results from monitoring tasks
3.1 Determine to whom the information should be communicated	Communication agreements Match between incidents and organizations Types of contacts Need for further collaboration
3.2 Determine the incidents worth mentioning	Communication agreements Match between incidents and organizations Priority order of incidents Possible future incidents

An agent-based solution for interconnecting entities is a promising option for implementing the monitoring task [9]. In this work, an agent-based modeling and simulation framework is used to capture critical infrastructure interdependencies. Rinaldi et al. [36] have presented a taxonomy of critical infrastructure interdependencies that covers the types of interdependencies, infrastructure environments, coupling and response behavior, infrastructure characteristics, types of failures and states of operations.

Altwood et al. [2] have presented a critical infrastructure response framework for smart cities. They recognize the importance of the Internet of Things (IoT) as an information source; sensor-actuator networks create the base network and information pertaining to a smart city is aggregated. Alcaraz and Lopez [1] have proposed a wide-area situational awareness framework for enabling situational awareness and threat analysis in distributed systems with a low human presence. Kopylec et al. [24] have developed a visualization engine for cascading incidents and expressing relations between physical and cyber entities. Most complex critical infrastructure environments incorporate multiple actors and teams; thus, the principal goal is to provide adequate situational awareness. Koskinen-Kannisto [25] has discussed the challenges to implementing effective situational awareness in collaborative environments with significant information sharing.

In the context of service-oriented architectures [18], some techniques have focused on multi-sector solutions that deal with situational awareness in adaptive coordination and service specifications [49,50]. Bagheri and Ghorbani [5] have presented a service-oriented architecture for critical infrastructures where services are placed in a dynamic layered model. Tolone et al. [44] have developed an agent-based approach within a brokered architecture to handle the daunting task of identifying vulnerabilities in interdependent infrastructures. Liu and Xi [28] have presented a technique based on copula theory that provides an algorithmic solution; they also consider physical, cyber, geographical and logical dependencies. Wang et al. [47] view critical infrastructures as complex systems with self-organizing characteristics. Zimmerman [51] and Zimmerman and Restrepo [52] have proposed an approach for cataloging critical dependencies for analyzing cascading effects in infrastructure sectors and reducing the negative consequences.

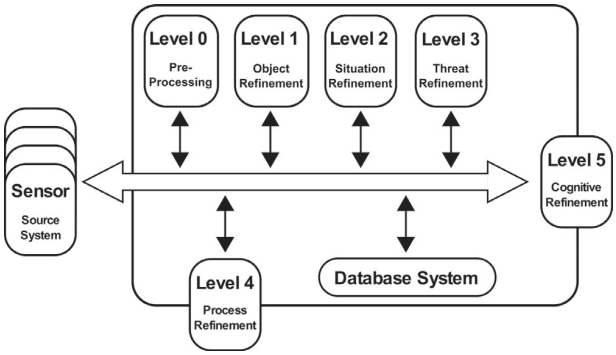


Fig. 3 – JDL data fusion model (adapted from [20,41]).

Jones and Endsley [21] have compared several measures (i.e., the SAGAT real-time probe measure, SART measure and NASA TLX workload measure) to evaluate the performance of an air sovereignty team in low- and high-workload tasks using the North American Aerospace Defense (NORAD) Regional Sector Air Operations Center simulator. Endsley and colleagues [16] have also compared the SAGAT and SART measures when assessing fighter pilot situational awareness. In other work, Endsley and colleagues [17] have compared the sensitivity and validity of SAGAT and SART when evaluating the situational awareness of air traffic controllers.

3. Information requirements for situational awareness

Situational awareness oriented design provides a means to develop optimized systems and software for situational

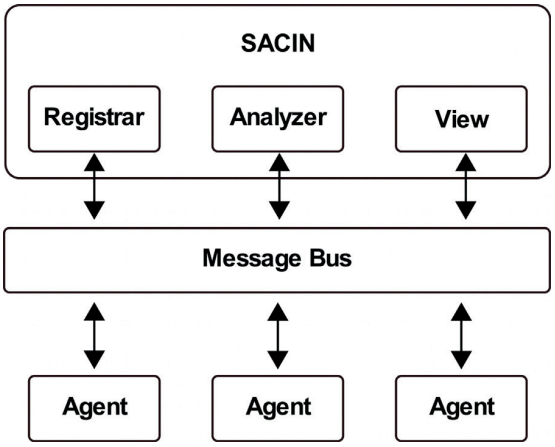


Fig. 4 – High-level representation of the main system components.

awareness and decision making applications [7,12,14,15]. Situational awareness oriented design is a three-phase methodology, which starts with task analysis to identify operator information requirements, followed by a design phase that creates a user-centered system based on the collected requirements. Finally, the system is tested using the target user group with the help of situational awareness measurement questionnaires and procedures. This process is repeated until the system meets the stipulated requirements. Situational awareness based design ensures that key information needed for situational awareness is passed to the complete operating picture system and is presented to operators in a manner that enhances their situational awareness measurably.

3.1. *Situational awareness requirements for critical infrastructure applications*

The situational awareness oriented design process was performed in two cycles, where the results from interviews and user tests were used to refine all the layers of the SACIN framework. Goal-directed task analysis, an essential part of the situational awareness oriented design process, was conducted during the research project. Goal-directed task analysis is a method for identifying the pieces of information that operators need to perform their tasks. It involves a series of semi-structured interviews [12] of subject matter experts who have extensive knowledge of operator tasks to create the basis for the situational awareness requirements. Goal-directed task analysis is used to discover the information that operators need to achieve the goals defined in the goal hierarchy [12].

Seven infrastructure sectors were covered in the research: (i)Power; (ii) water; (iii) information and telecommunications; (iv) banking and finance; (v) transportation; (vi) chemicals; and (vii) emergency services. The results of the interviews were used to create a goal hierarchy. The hierarchy comprises one major goal and three sub-goals, two of which have minor goals as shown in Fig. 2. The major goal for a critical infrastructure monitoring operator is to increase the situational awareness of the critical infrastructure. This high-level goal is divided into three main goals: (i) Monitor the environment; (ii) analyze incidents and communicate internally; and (iii) communicate externally [38].

After constructing the goal hierarchy and analyzing the interview data further, two distinctive blocks of information were identified. Table 1 describes the blocks; every mention of incidents and services or systems in the final requirements can be replaced with these blocks. The first type is incident-related information. This includes all the information that should be gathered during an incident. An incident in this case refers to a situation in which some fault, natural event or deliberate attack causes a warning or a fault in the target system [45]. The other type of distinctive information is system-related, which refers to system status, its purpose and service dependencies. Although the blocks contain the information needed for decision making, not everything needs to be present at all times.

Table 2 presents the complete situational awareness requirements. The requirements for the monitoring goal include ongoing incidents, services and systems affected by the incidents, their priority and possible future incidents. These requirements were all verified in the interviews: an operator needs to be aware of all the incidents and service status in the monitored environment in order to coordinate repairs and communicate the situation efficiently.

The situational awareness requirements for the second main goal include the elements needed in the monitoring task. However, an operator is also responsible for communicating with other individuals to create a common and shared situational awareness. The operator may also analyze an incident with other individuals.

The requirements related to external communications share similarities with the other two main goals. However, in the case of external communications, an operator must determine more carefully to whom he should forward

Table 3 – Key architectural components and their purpose.

Agent	Middleware component responsible for collecting data from the source systems; also handles connections to the SACIN back-end.
Registrar	Database component responsible for storing information about agents and the relationships between agents.
Analyzer	Component responsible for analyzing the impact of an event on the entire critical infrastructure system; includes a model of the critical infrastructure.
View	Component responsible for visualizing the critical infrastructure and the results provided by the analysis component.

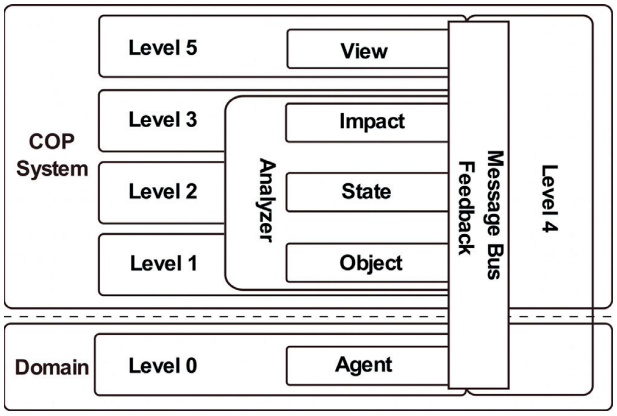


Fig. 5 – System architecture with the JDL data fusion model.

Fig. 6 – Display layout.

information; this requires the operator to know the communication agreements between organizations and be able to evaluate the types of contacts. Additionally, an operator may have to evaluate if further collaboration is required between organizations.

4. System architecture

The system architecture was designed to satisfy the requirements presented in the previous section. In order to construct an efficient architecture for the framework, the Joint Directors of Laboratories (JDL) data fusion model, a conceptual model suitable for information collection and data processing was selected. The model provides a conceptual framework for combining procedures and algorithms that refine sensor data in order to improve situational awareness. The original JDL sensor fusion model was presented in 1988 and enhanced in 1999 [41]. It has five levels, each of which refines and combines the available data from the previous levels to support advanced analyses and predictions. Fig. 3 shows the five levels of the JDL data fusion model.

Data fusion is often understood as a means to improve the prediction quality of ordinary physical sensors. As such, it is also suitable for cyber-physical sensors [20], which collect information from diverse sources that may not be accurate or reliable. Indeed, the notions of sensor fusion and noisy data form a natural analog. The sensors in a cyber-physical setting could, for example, be intrusion detection systems, or they could be host health monitoring products or network flow analyzers, or systems that log virtually any process. The outputs of these sensors can be combined and refined using, for example, vulnerability databases such as MITRE's Common Vulnerabilities and Exposures for further analyses.

The sensors in Fig. 3 correspond to various critical infrastructure information sources, such as intrusion detection systems, supervisory control and data acquisition (SCADA) systems and log files. The level 0 fusion process converts raw input data into a common format. Level 1 combines the preprocessed data and identifies objects such as systems, attacks and hardware malfunctions. The level 2 fusion process creates a system-level perspective from the current situation whereas the level 3 process predicts the future state of the system. The level 4 process manages the sensors and allows fusion process refinement. Level 5 serves as the interface between a human operator and the system.

The main components in the system architecture are the agents, registrar, analyzer and view. The components, in accordance with the JDL model, handle data collection, database management, analysis and user-interface-related tasks. Additionally, a common message bus is utilized to provide cross-component communications. The communications channel, which connects all the components to each other, can store events and messages independently.

Fig. 4 presents a high-level representation of the main system components and their relationships to each another. Table 3 lists the components and their purpose. Each component defines an interface that other components are able to use via the message bus. The registrar component requires a database in order to manage dependency information and user accounts. The analyzer component also requires a database for data fusion.

4.1. *JDL levels 0 and 1: Data collection*

Critical infrastructure source systems produce data and logs that must be translated into events for the common operating picture system. A system-specific component in JDL level 0 is required to collect, filter and transmit information from the source system to the common operating picture system. Since raw data cannot usually be collected and integrated due to volume, legal or security constraints, processing and integration are required for data collection at an early level. For example, log files for audit purposes may contain confidential user details and cannot be sent directly to an external system. Not only does this pose a security threat, but it is also not possible to configure the common operating picture system with knowledge about the minute details of every system that has to be integrated.

An agent-based client-server architecture, in which each source system is integrated via custom middleware/software, is a suitable solution for JDL level 0 integration. An agent col-

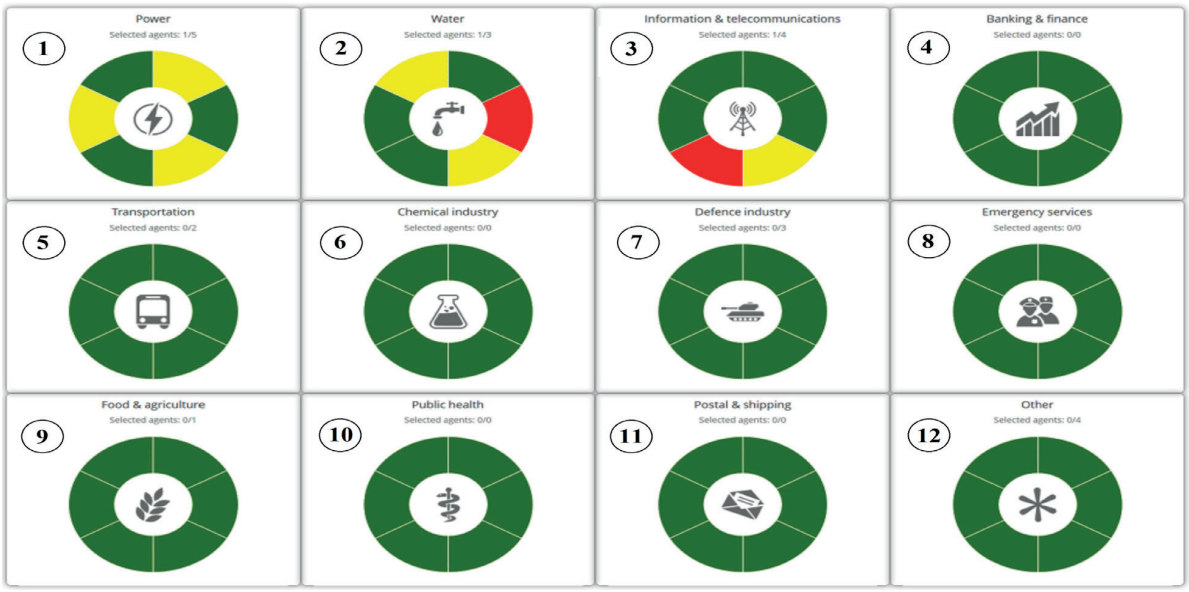


Fig. 7 – Overview display.

lects information from a source system and generates events that can be processed at a central location. Source system experts assist in customizing agents for their systems. Thus, diverse critical infrastructure source systems can be integrated without losing the system-specific knowledge provided by source system experts and without requiring full data capture.

SACIN agents comprise middleware/software and library components that facilitate centralized event logging and analysis. They enable critical infrastructure operators to interface source systems with the SACIN server. While a SACIN agent provides an interface for reporting and logging, it does not gather information about or analyze out-of-the-box events. Domain experts are required to articulate the logic for extracting essential data from their systems and interfacing with the agents. This approach enables the domain experts to define exactly what is sent to the SACIN system.

A SACIN agent is operated via a plug-in component (e.g., a vendor-specific intrusion detection log parser) that handles the interfacing between the source system and agent interface. Ideally, the plug-in should be developed by a domain expert who understands when the system is operating normally. The plug-in should be tailored to gather information from the system and analyze the current situation. When an anomaly is detected, the plug-in uses the interface provided by the agent to send a message to SACIN.

An agent generates events based on the data it receives from a source system. The events contain mandatory and optional key-value pairs that permit future extensions. Proper timing of the events is extremely important in a real-time situational awareness system. Because the system time could be inaccurate or incorrectly set, the SACIN agent incorporates a time service for external clock synchronization. The current implementation uses a Network Time Protocol (NTP) service to determine the time difference between a target system and the server. The time synchronization process runs as a separate thread every five minutes. This relatively short update

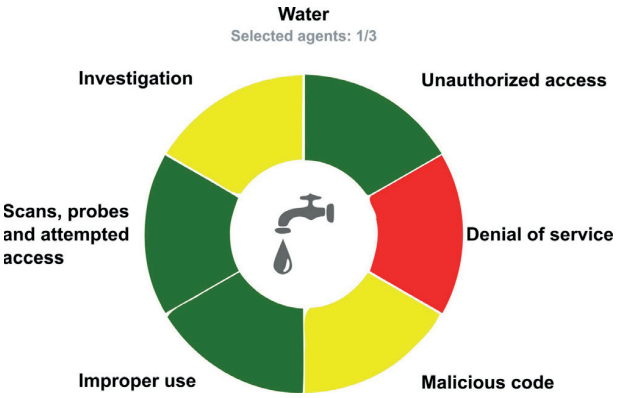


Fig. 8 – Status circle presented in the overview display.

interval is expected to mitigate fluctuation problems arising from unstable real-time clocks or sudden changes to the system time.

The agent middleware is written in Java and the Spring Framework [32]. The Apache Camel integration framework [4] and Apache Active MQ [3] implement communications with the SACIN server. The program is distributed as a single Java archive (JAR) that contains all the required dependencies.

The system architecture was tested using real-world data from intrusion detection system logs and a SCADA system snapshot. Custom agents for intrusion detection and SCADA systems enable the system to process event and dependency data from diverse sources to create a common operating picture and, thus, support situational awareness [26].

4.2. *JDL levels 2 and 3: Modeling and analysis*

The core task of the common operating picture system is to analyze agent-generated event streams and construct the common operating picture. In order to satisfy the requirements for JDL levels 2 and 3, an appropriate model of a critical infrastructure that is capable of handling a large amount of information is necessary. Due to the complexity and size of a critical infrastructure, the model must be highly specific to situational awareness tasks and only essential knowledge about the operational status of the systems should be considered. For these reasons, it was decided to omit all domain-specific details (e.g., electricity flows and other technical aspects) in order to simplify the model and make it suitable for national-scale deployment – running a heavy, detailed model at the national scale is computationally prohibitive.

The model requirements were outlined as described in Section 3. The model must present interdependencies, provide impact estimates of incoming events and deliver estimates of future status. The following list details the model requirements as defined by Puuska [33]:

- The location of each system must be provided; this could be specified in terms of geographical coordinates or via some other practical approach.

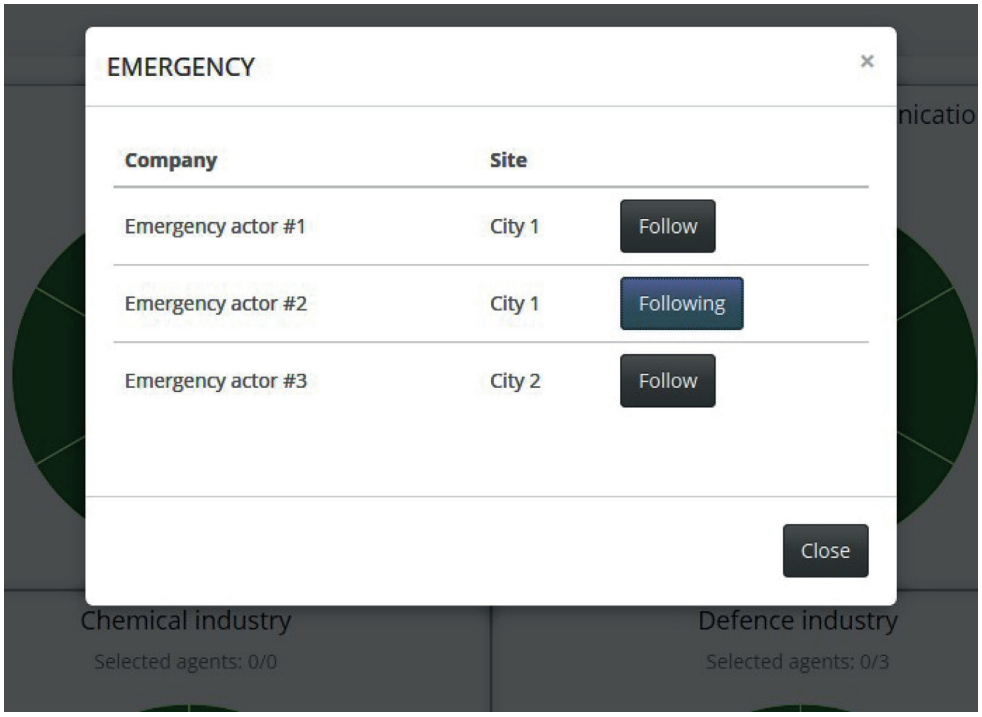


Fig. 9 – Filtering option.

- Every system must have an operational status; each system must have a tag that conveys its current or best known status.
- Each system must have a criticality or priority value; a metric should be used to rank systems based on their relative importance in the critical infrastructure.
- Dependencies between systems must be, at least partially, known; if the proper operation of a system is dependent on some other system, then this relationship must be modeled.
- The sufficiency of critical resources must be modeled; if a resource (e.g., backup power) is stored in a system, then the depletion process must be modeled.

The following information is needed to estimate the critical infrastructure status and future developments [33]:

- The location of each event must be specified; this could be specified in terms of geographical coordinates or via some other practical approach.
- Magnitudes must be quantifiable at the system and critical infrastructure levels.
- Relations between incidents and systems must be specified.
- Durations must be quantifiable at the system and critical infrastructure levels.

A dependency-heavy formalism was selected to handle the interconnectedness of critical infrastructures. Critical infrastructures can be modeled as a dependency graph, where the nodes represent critical infrastructure entities and the edges represent dependency relationships between them. The operational status of an entity was modeled by combining graphs and finite state machines. Each node is associated with a finite state machine, which represents the state (health, capability, etc.) of the entity in question. Interested readers are referred to [34] for details about this approach.

As time passes, it is increasingly likely that the current status of an entity would not reflect the latest sensor reading; moreover, the uncertainty about the state of the sensor increases. The probabilities of previously observed or known behavior of the entity can be used to make predictions about the current state. A probability distribution may obtained by observing the operation of the sensor over a long period of time or it could be specified by the sensor operator [22,23].

The proposed model was tested using real-world data, which included a dataset gathered during a large-scale storm. The telecommunications network and electric grid of the coastal area of Finland were modeled using the graph model. The results indicated that the model satisfies the stipulated requirements [33].

4.3. *JDL level 4: Message transport, feedback and database components*

The JDL data fusion model shown in Fig. 3 requires all the data fusion processes to work together. The inter-component communications channel is a key feature that facilitates operation in a distributed environment and in a national-scale implementation. As shown in Fig. 4, a suitable communications channel can be implemented via a common message bus. The common message bus must handle large numbers of messages from multiple sources and route messages to one or more destinations. Additionally, it must support distributed deployment across multiple servers for scalability and fault tolerance. Fig. 5 shows the architecture of the common operating picture system, which adheres to the JDL model and accommodates all the data fusion sub-processes.

A brokered architecture is a suitable solution for the distributed environment. Brokering can be viewed as a cloud service in which a group of servers collectively offer message transfer services. Services such as broadcasting and bidirectional messaging can be offered with little overhead. Additionally, because most of the communications between components are of the "fire-and-forget" type, the architecture readily scales to handle all the communications between diverse system components.

The main task of the registrar (Fig. 4) is to register agents with the common operating picture system. The registrar also provides user authentication and authorization services for the entire system. Source system experts register agents and define their dependencies. When an agent is registered, it is assigned a unique identifier in order to associate events with the source system. The registrar also provides a means for system experts to enter and maintain their agent information (including dependencies) in other systems.

4.4. *JDL level 5: View and user interface*

A view component handles data presentation via a user interface. It is a key part of the chain that transforms raw data to human operator situational awareness.

The SACIN user interface is implemented via a web application that executes on computers in a control center, computer workstations, laptops or mobile phones. Access via a web browser makes the user interface customizable and eliminates the need for software to be installed on an end-user device.

The implemented user interface provides four distinct views, each displaying event data received from the SACIN analysis component. Fig. 6 shows the display layout. The layout presents the most relevant information in the middle portion of the display, where an operator looks at most of the time. The other displays could be removed and all the data could be presented in tabs on a single display.

Each of the four views serves a different purpose: (i) Display a general overview of the monitored environment; (ii) display the geographical distribution of events; (iii) display the logical relationships between actors; and (iv) display the temporal distribution of events. Individual visual elements include status circles, a timeline, raw event log, geographical map and logical map. Because the displayed data is diverse, multiple display monitors are required instead of a single monitor [48]. Interactions are implemented using JavaScript; ten open-source libraries were used to create the various views.

The SACIN user interface was originally developed by Rummukainen et al. [37]. The original version has been enhanced to provide a better logical display and to improve usability.

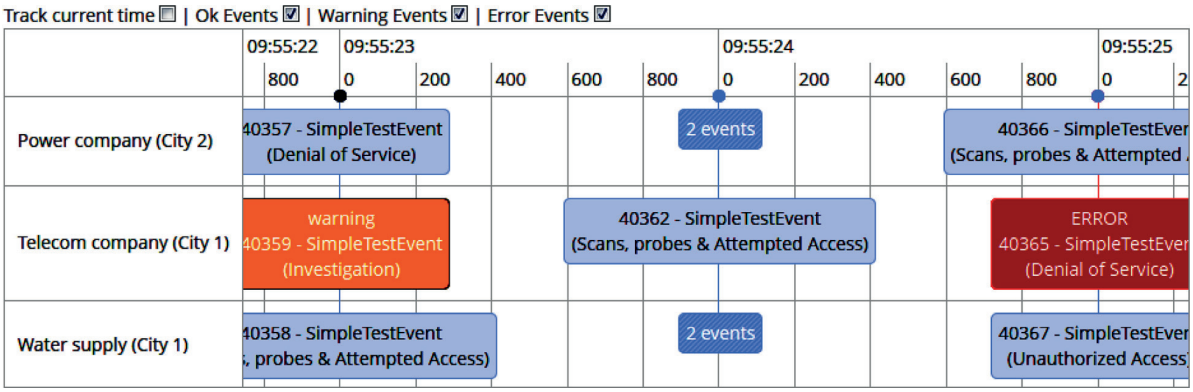


Fig. 10 – Timeline for visualizing the event stream.

Show 10 entries Search:

Id	Event	Agent	Severity	Category	Occurred	Check	Hide
40365	SimpleTestEvent	Telecom company (City 1)	9	Denial of Service	29. marraskuuta 2013 9:55:25		Hide
40366	SimpleTestEvent	Power company (City 2)	3	Scans, probes & Attempted Access	29. marraskuuta 2013 9:55:25		Hide
40367	SimpleTestEvent	Water supply (City 1)	0	Unauthorized Access	29. marraskuuta 2013 9:55:25		Hide
40360	SimpleTestEvent	Power company	4	Improper Use	29. marraskuuta 2013 9:55:24		Hide

Fig. 11 – Event log for tracking all the data retrieved from SACIN.

4.4.1. *Overview display*

The overview display is designed to enable an operator to quickly check that all the systems in the monitored critical infrastructure are working as they should. The display delivers information that satisfies the goal-directed task analysis requirement 1.2: monitor resource management (see Fig. 2). The display layout shown in Fig. 7 has twelve status circles. The critical infrastructure categorization of Lewis [27] was engaged to develop the overview display sectors. An extra sector was added for entities that do not belong to the other categories.

Support for global situational awareness is vital. As a result, the display provides an overview of the situation across an operator's goals at all times (with detailed information about the current goals) and enables efficient and timely goal switching and projection (situational awareness oriented design principles 4 and 5, as described in [12]). Critical information that indicates that the goal should be switched is rendered salient by implementing situational awareness oriented design principle 6, as described in [12].

Each critical infrastructure sector icon has a status circle around it. The individual status circle shown in Fig. 8 is partitioned into six segments that represent U.S. federal agency incident categories [45]. Thus, a human operator can obtain an overall understanding of the types of events occurring in the various industry sectors. This is especially important for operators who are just beginning their shifts because they may not have prior knowledge about the state of the critical infrastructure.

Visual cues convey information about important events in the critical infrastructure. The classic traffic light analog is used to display whether or not critical infrastructure systems are working adequately, if they are having some difficulties or if their services are severely degraded. Gray-scale industry icons are also used to enable operators to identify industries easily, without interfering with the event colors in the status circle. Although the incident category names are not displayed in the user interface to conserve display space, category names are revealed when an operator hovers over a segment with a mouse.

An operator may not always wish to monitor the entire critical infrastructure. In such a situation, an operator can use the filtering option illustrated in Fig. 9 to click on industry icons and select the specific systems to be monitored. In the exam-

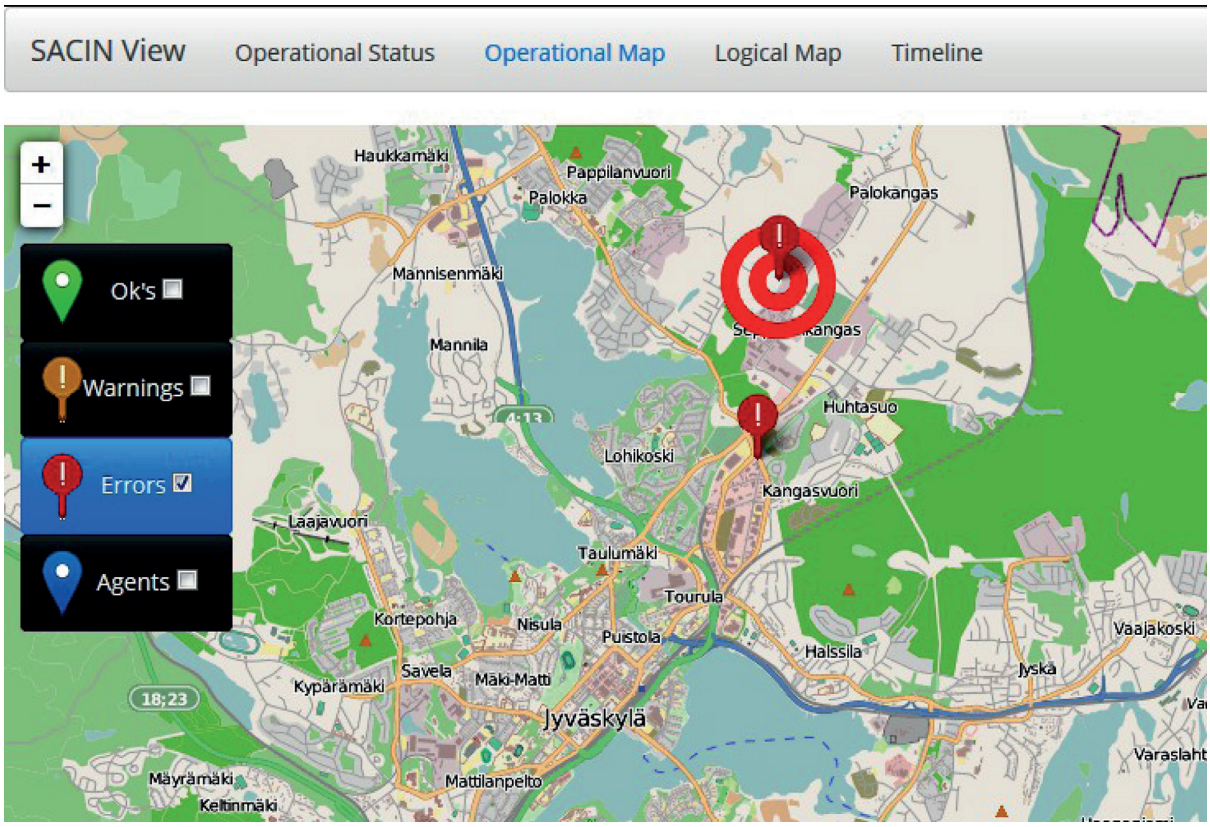


Fig. 12 – Map display.

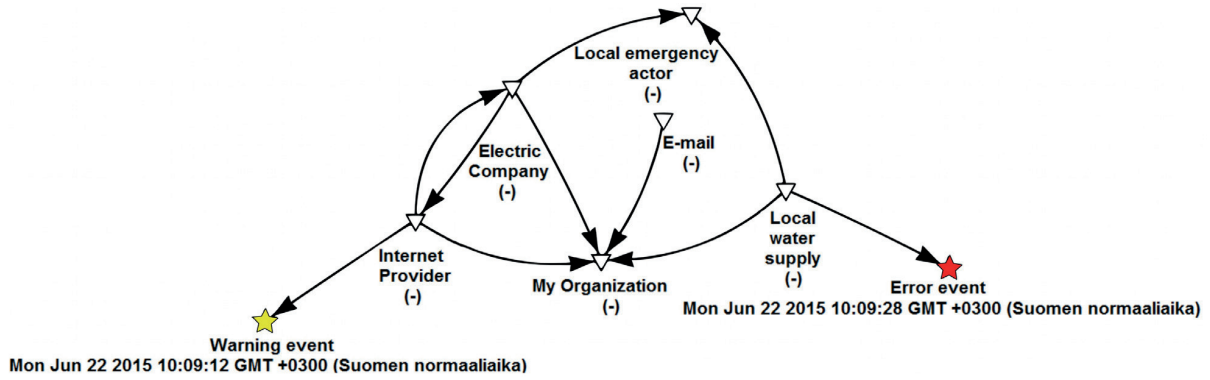


Fig. 13 – Logical display.

ple, the emergency services icon (rightmost icon in the middle row of Fig. 7) has been clicked and the operator can select the systems of interest.

4.4.2. *Temporal display*

Operators need to monitor events that have occurred and their temporal characteristics. For this reason, one of the displays is dedicated to a timeline and event log, which are illustrated in Figs. 10 and 11, respectively. This enables an operator to have a temporal perspective of the critical infrastructure and to link events, even when there are no indications of connections between the events in the other displays or external sources. The temporal display provides the information needed to satisfy the goal-directed task analysis goals 1.1 and 2 (see Fig. 2) and to handle major decisions associated with these goals.

When the user interface receives new events, the timeline displays a new sequence line for every new critical infrastructure entity. Each entity is placed on a separate line to pre-

Table 4 – Correlations between SUS and the situational awareness measures.

	SART	SART understanding	SART supply	SART demand	SAGAT	SAGAT level 1	SAGAT level 2	SAGAT level 3	Performance
SUS									
Pearson correlation	0.128	0.144	0.173	0.056	0.053	0.043	−0.107	0.032	0.288
Sig. (Two-tailed)	0.533	0.484	0.397	0.785	0.796	0.835	0.604	0.877	0.153

Table 5 – Overall correlations between SART, SAGAT and performance.

		SART	SART understanding	SART supply	SART demand	Performance
SAGAT	Pearson correlation	0.345	0.257	0.227	−0.181	0.314
	Sig. (Two-tailed)	0.084	0.206	0.265	0.375	0.119
SAGAT	Pearson correlation	0.248	0.160	0.138	−0.176	0.320
Level 1	Sig. (Two-tailed)	0.223	0.436	0.502	0.390	0.111
SAGAT	Pearson correlation	0.378	0.353	0.058	−0.283	0.138
Level 2	Sig. (Two-tailed)	0.057	0.077	0.777	0.160	0.503
SAGAT	Pearson correlation	0.062	−0.157	0.348	0.015	0.051
Level 3	Sig. (Two-tailed)	0.763	0.444	0.082	0.941	0.803
Performance	Pearson correlation	0.415*	0.011	0.397*	−0.419*	1
	Sig. (Two-tailed)	0.035	0.958	0.044	0.033	

*Correlation is significant at the 0.05 level (two-tailed test).

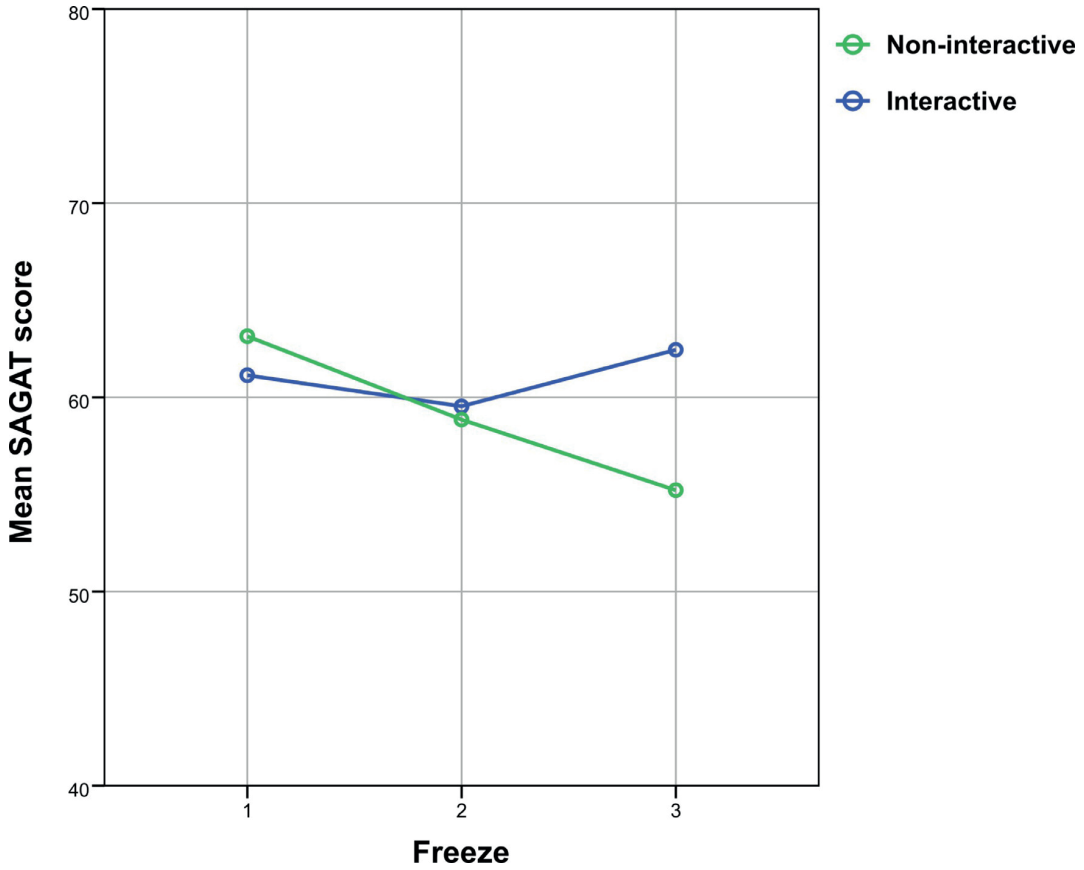


Fig. 14 – SAGAT scores at each freeze in the first scenario.

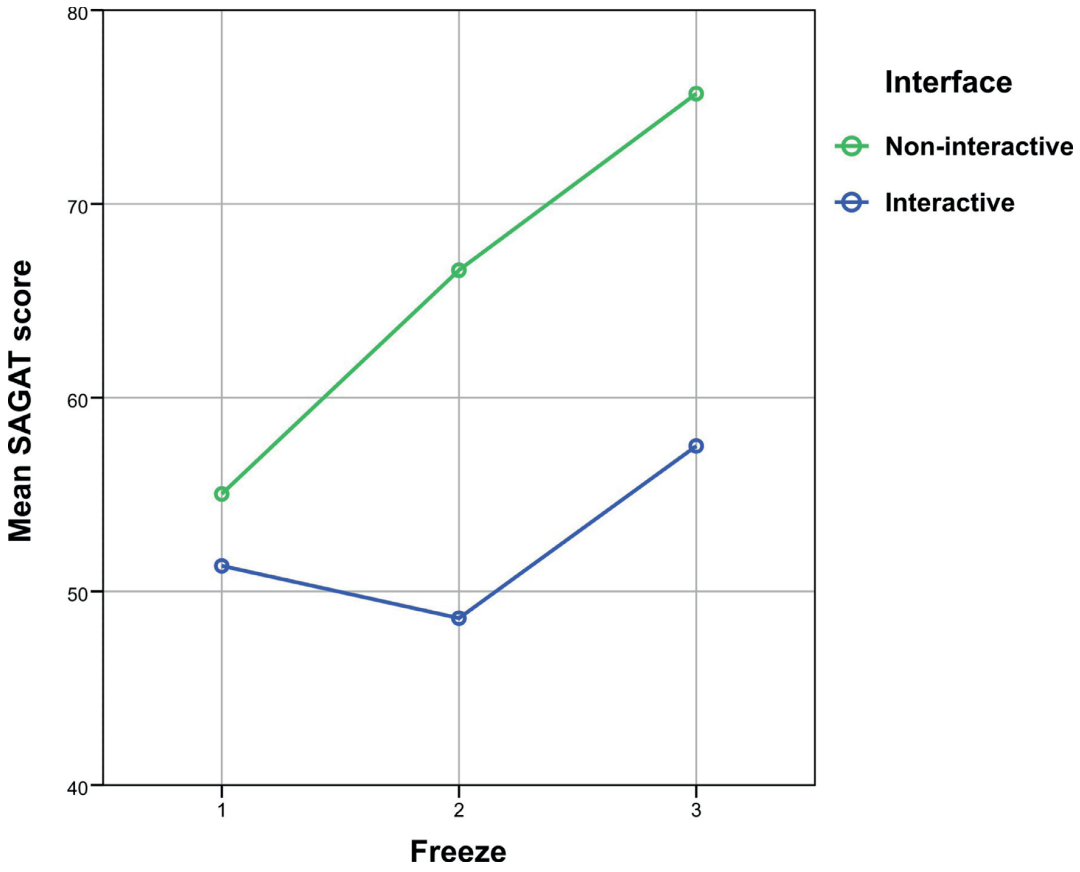


Fig. 15 – SAGAT scores at each freeze in the second scenario.

vent confusion. The timeline uses a slightly different traffic light color analog for events, replacing the color green with light blue. This is intended to increase the ability of an operator to spot warnings and alerts as and when they occur. The timeline is also scalable and helps prevent information overload. For example, as shown Fig. 10, events that occur close to each other are grouped together and are labeled with the number of events. The color of the group element corresponds to the most severe event it contains. The operator can then zoom in, pan or follow the current time on the timeline as desired.

New events are also displayed immediately in the common event log next to the timeline. The event log displays all the raw data of the event and also has hide and check buttons to promote operator interaction. An event may also be highlighted by clicking on a specific row, which displays the event in the other views. This enables the operator to link multiple monitors [48].

4.4.3. *Map display*

One aspect of situational awareness in the critical infrastructure context is awareness of the geographical distribution of events and entities. The idea was to implement a map interface that would be easy to learn and operate. Thus, a common interface type that resembles popular map interfaces such as Google Maps was employed. This display provides the information needed for goal-directed task analysis goal 1.1 (see Fig. 2). Fig. 12 shows the map display in which events and actors are presented using different symbols.

New events are displayed as markers on the map; a marker can be clicked to show more information about the event. The traffic light analog is used to indicate event severity. Entities are displayed with blue markers. After new events are displayed on the map, they are also highlighted, as shown in Fig. 12. An operator is free to choose the types of markers displayed on the map. In the example, the operator has hidden all the markers except for the error markers.

The map interface also has a clustering feature that is intended to prevent information overload (situational awareness oriented design principle 46 [12]). Events are grouped based on their severity and an operator has the option to zoom in on an event to view the distribution of events. Also, although it is not shown in Fig. 12, the map interface can display areal events. Specifically, a colored area is displayed on the map that works in the same way as an event marker.

4.4.4. *Logical map display*

A key goal of SACIN is to support the analysis of dependencies between critical infrastructure systems [43], a feature that facilitates risk analysis. For this reason, one of the displays supports the visualization of the logical dependencies between critical infrastructure entities. This display presents system

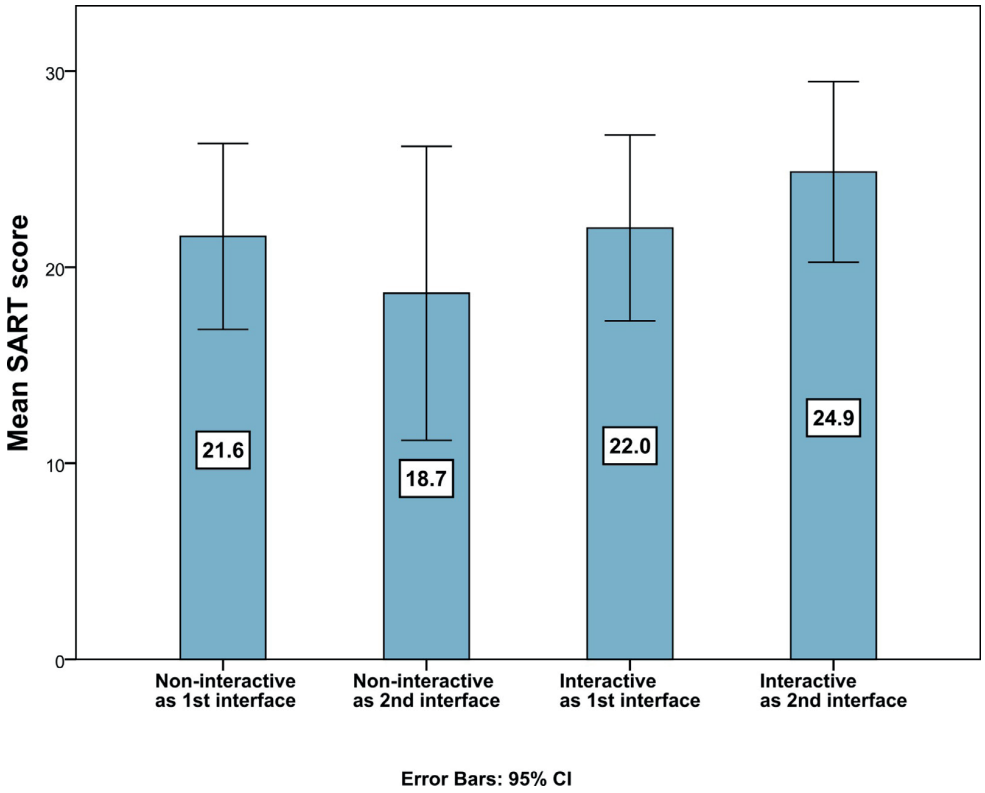


Fig. 16 – SART survey results.

dependencies in the form of a directed graph, enabling an operator to identify critical paths and estimate the amount of time it would take for negative events to propagate.

Fig. 13 shows the logical display with a directed graph implementation that enables an operator to evaluate event impacts across the monitored critical infrastructure environment. Systems and events are displayed as user-defined symbols, triangles and color-coded stars, respectively. The arrows in the graph illustrate service dependencies; the entity at the pointed end of an arrow is the dependent entity and the value near the arrow indicates how long the dependent entity can function normally without the service. The display presents level 2 information directly and provides assistance for level 3 projections (situational awareness oriented design principles 2 and 3 [12]).

Two variants of the logical display have been implemented: (i) Non-interactive interface; and (ii) interactive interface. This has made it possible to evaluate the impact of interactivity on system usability and situational awareness. The interactivity supported by the logical display enables an operator to create new nodes and dependencies, facilitating the incorporation of physical and abstract concepts in the critical infrastructure environment.

5. Evaluation and results

This section describes the evaluation methodology and the results.

5.1. *Evaluation methods*

Six methods were used to evaluate the SACIN user interface: (i) Situation Awareness Global Assessment Technique (SAGAT) [11]; (ii) Situation Awareness Rating Technique (SART) [42]; (iii) System Usability Scale (SUS) [8]; (iv) visual walkthrough [30]; (v) informal walkthrough [35]; and (vi) eye-tracking.

The user evaluations involved two iterations. The first iteration is described in [37]. The second iteration, which is described in detail in this paper, involved SAGAT, SART and SUS testing. In addition, objective performance measures were collected. Since all the participants in the first iteration had experience in monitoring critical infrastructure environments, the second iteration focused on evaluating whether an inexperienced user could become acquainted with the monitoring system with little or no prior knowledge.

A total of thirteen individuals participated in the second iteration. The mean age of the participants was 36.7 years (SD = 1.50). The voluntary participants were male graduate students who were attending the General Staff Officer Course at National Defence University (Helsinki, Finland). Prior to the evaluation, all the participants received instructions on SAGAT, SART and SUS.

Two 20-minute scenarios were employed. The system was evaluated with two interfaces, the non-interactive interface and the interactive interface, as mentioned in the discussion of the map display. The scenarios represented common situ-

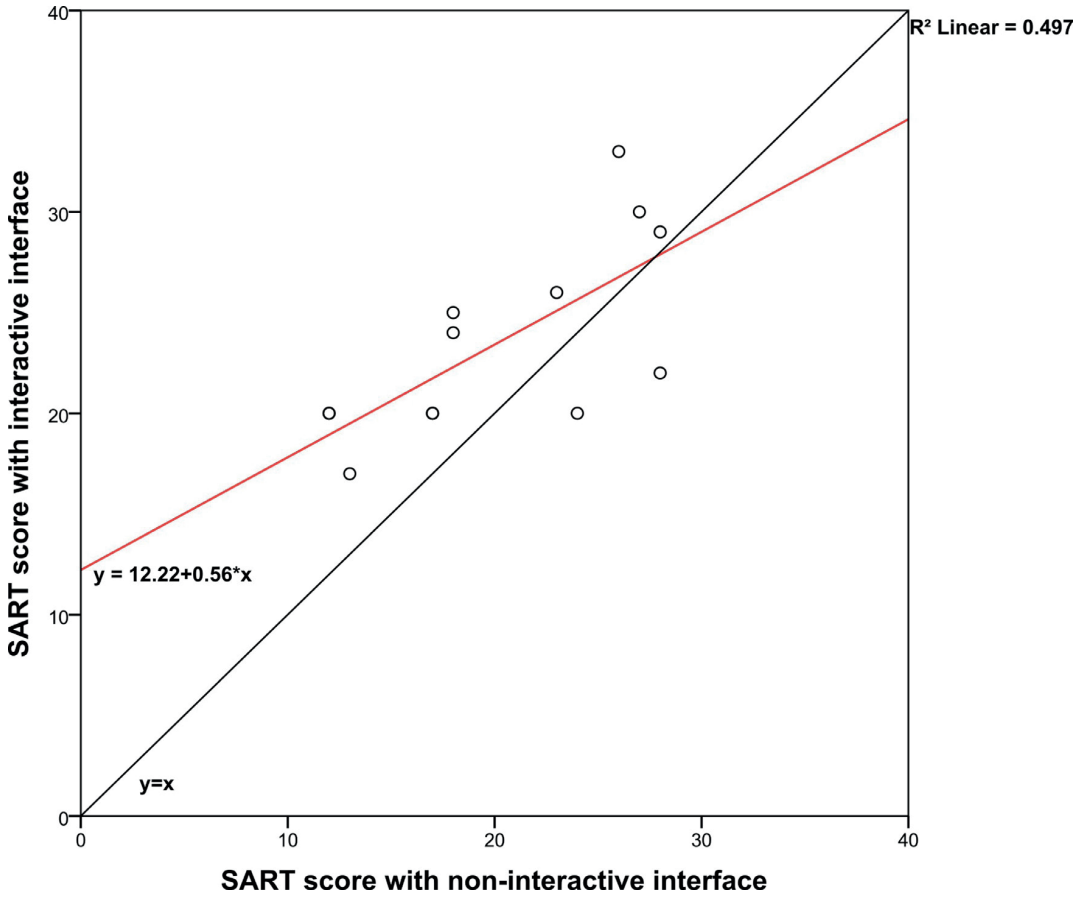


Fig. 17 – SART survey results.

ations that operators would face in their daily work. The first scenario involved the loss of power in the "environment" area, which required operator action. The second scenario did not focus on a specific incident, but on normal operator activities.

In each scenario, the participants had to monitor a specific entity and evaluate the threats it faced. The performance of the participants was measured via the described test batteries. Each participant was instructed to notify a supervisor (seated behind him) if the monitored entity was perceived to be threatened.

5.1.1. *SAGAT*

The Situation Awareness Global Assessment Technique (SAGAT) [11] was used to evaluate the level of situational awareness provided by the user interface. The evaluation required the same scenarios to be presented to all the participants. Answers were scored based on the scenario outcomes.

SAGAT uses queries designed to assess the actual participant situational awareness, including level 1 (perception of the elements), level 2 (comprehension of their meaning) and level 3 (projection of future status). In the SAGAT tests, the situational awareness requirements for a critical infrastructure operator [38] were used to generate questions for SAGAT freezes during the scenarios. Three SAGAT freezes were inserted at random times during each scenario and the responses were recorded manually (i.e., using a pencil and paper).

5.1.2. *SART*

The Situation Awareness Rating Technique (SART) [42] was used to evaluate the subjective level of situational awareness. The SART surveys comprised ten statements that were evaluated on a seven-point scale. The surveys were administered after each scenario. A total of 26 SART surveys were completed – one survey by each participant for each scenario.

5.1.3. *SUS*

The ten-point System Usability Scale (SUS) is designed to evaluate system usability [8]. In the research, SUS was used to evaluate the ease of use of the system implementation. SUS surveys were filled out alongside the SART survey; thus, there were also 26 completed SUS surveys.

5.2. *User evaluation results*

The Student's *t*-test and Pearson correlation were used in the statistical analysis. Performance was scored on a scale from 1 (worst) to 3 (best). The mean value with standard error for performance was 2.50 ± 0.11 (with standard deviation (SD) = 0.58).

5.2.1. SAGAT results

The SAGAT questions were divided into three groups according to the level of situational awareness they represented. The SAGAT scores (total and specific levels) were expressed as percentages of the highest possible score (42, 36, 10 and 88 for levels 1, 2, 3 and total, respectively). Thus, the scale for each score ranged from 0 to 100.

In the evaluation, the mean total SAGAT score with standard error was 59.3 ± 1.89 (SD = 9.7), with the scores ranging from 40.0 to 82.1. The mean scores with standard errors were 57.1 ± 2.93 (SD = 15.0) for level 1 situational awareness, 66.6 ± 1.96 (SD = 10.0) for level 2 and 53.2 ± 4.36 (SD = 22.2) for level 3. Notably, the SAGAT scores for level 2 were significantly higher than the scores for levels 1 and 3 ($p < 0.01$). This is because the situational awareness oriented design principles are focused on user interfaces that can provide high levels of situational awareness to operators. As such, level 1 would not provide an operator with a high level of situational awareness [13].

5.2.2. SART results

The overall SART score for situational awareness was computed using the formula

$$SA = U - (D - S)$$

where U is the understanding of the situation provided; D is the amount of demand on attentional resources; and S is the supply of attentional resources (perceived workload).

The scales for U, D and S ranged from 4 to 28, 3 to 21 and 3 to 21, respectively. Thus, the overall score for SART ranged from −14 to 46. The mean overall SART score with standard error was 22.0 ± 1.12 (SD = 5.7), with the scores ranging from 12 to 33. The mean scores with standard errors were 12.73 ± 0.62 (SD = 3.2) for understanding, 20.69 ± 0.51 (SD = 2.6) for supply and 11.38 ± 0.61 (SD = 3.1) for demand.

5.2.3. SUS results

The scale for SUS scores ranged from 0 to 100. A score above 68 is usually considered to be above average [40]. In the evaluation, the mean score with standard error for SUS was 77.4 ± 2.39 (SD = 12.2), which is greater than the average score mentioned above. The scores ranged from a minimum of 52.5 to a maximum of 95.

5.2.4. Correlations between situational awareness measures and SUS

A Pearson correlation matrix was computed to directly compare the measures employed in the study. Table 4 presents the correlations between SUS and the various situational awareness measures. The results reveal that no significant correlation exists between SUS and SART, SUS and SAGAT, and SUS and performance. For example, the correlation between SUS and performance was determined to be (0.288, $p = 0.153$), which implies that no correlation exists.

Table 5 presents the overall correlations between SART, SAGAT and performance. Previous studies have shown that SART scores do not correlate with SAGAT scores [16], which was also mostly the case in this research. However, weak correlations may exist between SAGAT and SART (0.345, $p = 0.084$), SAGAT level 2 and SART (0.378, $p = 0.057$), SAGAT level 2 and SART understanding (0.353, $p = 0.077$), and SAGAT level 3 and SART supply (0.348, $p = 0.082$). Some significant, albeit moderate, correlations exist between performance and various SART categories: overall (0.415, $p = 0.035$), supply (0.397, $p = 0.044$) and demand (−0.419, $p = 0.033$). A weak correlation between SAGAT and SART was also observed in [46], but it was deemed to be non-significant in [16,21].

In summary, the SUS, SART and SAGAT scores do not have significant correlations with each other. This is an important finding because the SUS score attempts to represent the usability of the evaluated system. However, if the SUS score does not correlate with the SAGAT and SART scores, then an operator may believe that the system has good usability although it does not necessarily provide the required situational awareness. Thus, the results indicate that, when designing a user interface for a monitoring system, the interface should not be evaluated solely on the basis of situational awareness or usability. Instead, both aspects must be taken into account in order to design a system that offers good situational awareness and usability. Salmon et al. [39] suggest that, in order to fully assess situational awareness, a battery of compatible, but different, measures, including SART and SAGAT, should be employed. Since SUS does not correlate with the other measures in this research, it is necessary to add SUS to the battery of measures.

5.2.5. Impacts of scenarios and interfaces

The mean values with standard error for performance were 2.31 ± 0.18 (SD = 0.63) for the non-interactive interface and 2.69 ± 0.13 (SD = 0.48) for the interactive interface. The performance with the non-interactive interface appears to be better than the performance with the interactive interface, although the difference is not significant ($p = 0.054$).

The SAGAT scores presented in Figs. 14 and 15 show the development of the situational awareness goal in the evaluation scenarios. As noted above, the participants in the evaluation employed non-interactive and interactive interfaces. Each score corresponds to the total percentage of correct answers to SAGAT queries at a simulation freeze. The trend line represents the total SAGAT score, which comprises the three levels of situational awareness.

The SAGAT scores between the alternatives are inconclusive. In the second scenario, the interactive interface appears to provide better situational awareness toward the end of the scenario, but the only statistically significant difference is at freeze 2 ($p = 0.027$).

Fig. 16 shows the SART survey results. The participants above the line $y = x$ scored better with the interactive interface. A Pearson correlation of 0.705 ($p = 0.007$) exists between the results. The red linear regression line ($R^2 = 0.497$) demonstrates that the interactive interface gives better results ($p = 0.17$), especially when the participants have low situational awareness.

Fig. 17 presents some additional SART survey results. The results indicate that the participants believed that their situational awareness increased by an average of 3.29 points when they switched from the non-interactive interface to the interactive interface. Also, the participants felt that their situational awareness fell by an average of 3.33 points when

Table 6 – Correlations between situational awareness measures and SUS for the two user interfaces.

Interface		SUS	SART	SART understanding	SART supply	SART demand	Performance
Non-interactive interface							
SUS	Pearson correlation	1	0.133	0.092	0.153	−0.055	0.299
	Sig. (Two-tailed)		0.666	0.764	0.617	0.858	0.321
SAGAT	Pearson correlation	−0.161	0.230	0.330	0.200	−0.045	−0.110
	Sig. (Two-tailed)	0.600	0.449	0.272	0.512	0.883	0.720
SAGAT	Pearson correlation	−0.177	0.073	0.237	0.165	0.156	0.010
Level 1	Sig. (Two-tailed)	0.564	0.812	0.435	0.590	0.611	0.974
SAGAT	Pearson correlation	−0.253	0.191	0.310	−0.177	−0.307	−0.196
Level 2	Sig. (Two-tailed)	0.404	0.531	0.302	0.564	0.307	0.522
SAGAT	Pearson correlation	0.115	0.315	0.116	0.505	−0.077	0.109
Level 3	Sig. (Two-tailed)	0.709	0.294	0.705	0.078	0.802	0.723
Performance	Pearson correlation	0.299	0.437	−0.306	0.592*	−0.493	1
	Sig. (Two-tailed)	0.321	0.135	0.309	0.033	0.087	
Interactive interface							
SUS	Pearson correlation	1	−0.025	0.117	0.116	0.290	0.124
	Sig. (Two-tailed)		0.936	0.704	0.707	0.336	0.687
SAGAT	Pearson correlation	0.175	0.376	0.149	0.175	−0.291	0.682*
	Sig. (Two-tailed)	0.567	0.205	0.627	0.568	0.335	0.010
SAGAT	Pearson correlation	0.196	0.365	0.065	0.042	−0.489	0.585*
Level 1	Sig. (Two-tailed)	0.521	0.221	0.832	0.892	0.090	0.036
SAGAT	Pearson correlation	−0.138	0.513	0.328	0.270	−0.203	0.364
Level 2	Sig. (Two-tailed)	0.652	0.073	0.274	0.372	0.505	0.221
SAGAT	Pearson correlation	−0.098	−0.340	−0.371	0.111	0.150	−0.033
Level 3	Sig. (Two-tailed)	0.751	0.256	0.212	0.718	0.624	0.915
Performance	Pearson correlation	0.124	0.222	0.082	−0.030	−0.284	1
	Sig. (Two-tailed)	0.687	0.466	0.790	0.923	−0.284	

*Correlation is significant at the 0.05 level (two-tailed test).

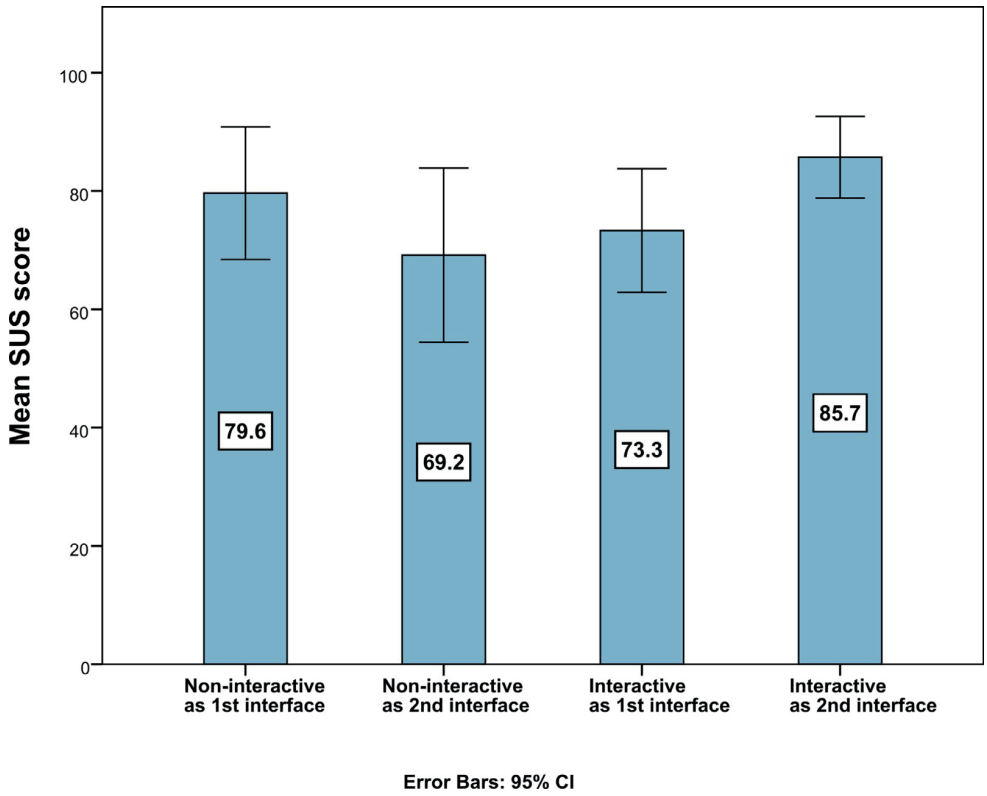


Fig. 18 – SUS survey results.

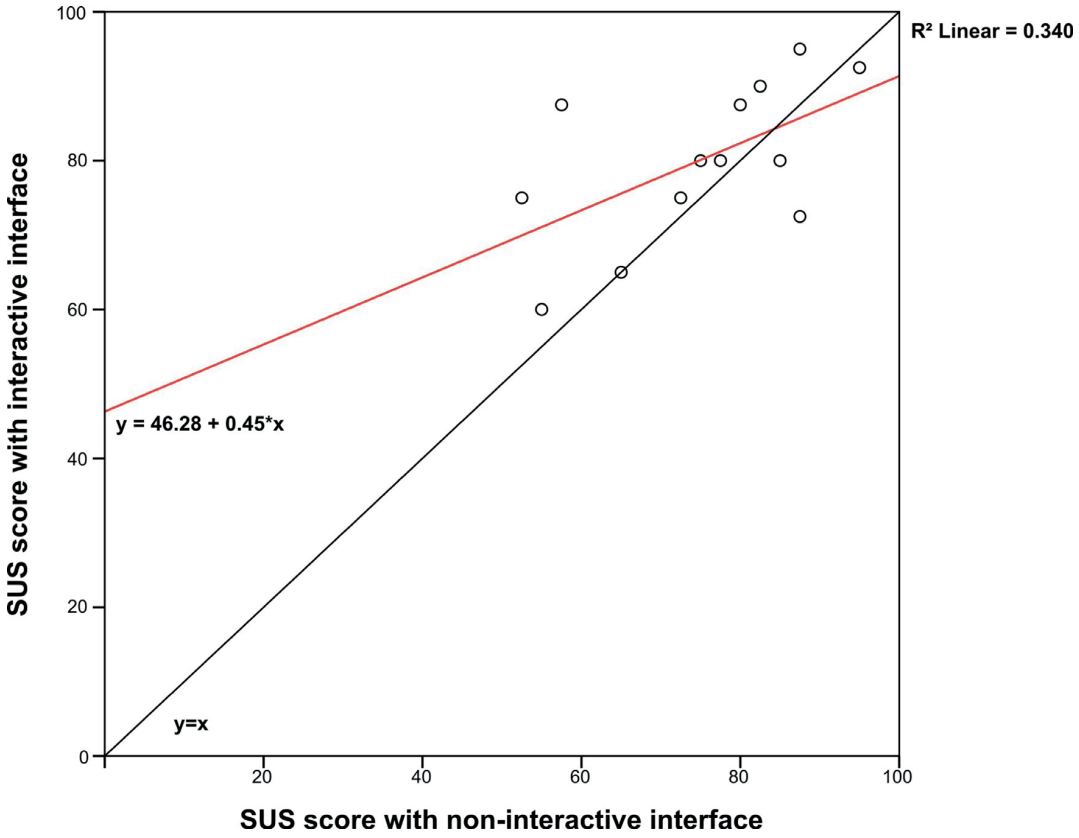


Fig. 19 – Scatter plot of the SUS survey results.

switching from the non-interactive interface to the interactive interface.

The mean values of the overall SART scores with standard errors were 20.23 ± 1.68 for the non-interactive interface and 23.54 ± 1.33 for the interactive system. The overall SART results between the two interfaces have a statistically significant difference ($p = 0.017$). This means that the participants felt more confident about their situational awareness when interacting with the logical display. As seen in Fig. 17, the participants tended to have better overall SART scores with the interactive interface, especially if they had low scores with the non-interactive interface. The results imply that the participants felt more confident when using the interactive interface. In a real-world scenario, this can be beneficial as well as harmful. This is because the correlation between SART and performance in the case of the interactive interface is not significant ($p = 0.466$); this will be discussed below in the context of the results presented in Table 6. Figs. 16 and 17 also show the impact of the interface evaluation order.

The mean values of the overall SUS scores with standard errors were $74.81. \pm 3.77$ for the non-interactive interface and 80.00 ± 2.91 for the interactive interface. Although the interactive interface scored better, no statistically significant difference exists between the two interfaces ($p = 0.125$).

Fig. 18 shows the SUS survey results. As in the case of the SART results, when the participants moved from the interactive interface to the non-interactive interface, their SUS scores fell by an average of 4.1. On the other hand, when the participants moved from the non-interactive interface to the interactive interface, their SUS scores increased by an average of 6.1.

Fig. 19 shows a scatter plot of the SART survey results. Participants above the line scored better with the interactive interface than the non-interactive interface. The difference in the mean values is not statistically significant ($p = 0.125$).

The results, which are supported by feedback provided by the participants, suggest that the SUS results are similar to the SART results. In particular, the participants with relatively low overall scores tended to score better with the interactive interface than the non-interactive interface compared with the participants with relatively high overall scores. However, it is notable that a relatively large statistical dispersion exists in the SUS scores. Figs. 18 and 19 also show the impact of the interface evaluation order.

Table 6 presents the interface-specific correlations. In the case of the non-interactive interface, the only significant correlation exists between the supply of attentional resources (SART supply) and performance (0.592, $p = 0.033$). This result is quite intuitive – the greater the attentional resources that remain, the better the performance.

In the case of the interactive interface, significant correlations exist between the SAGAT level 1 score and performance (0.585, $p = 0.036$), and between the overall SAGAT score and performance (0.682, $p = 0.010$). The first correlation could be

due to the nature of the performance measure, which is based on the perception of different elements and is measured by the SAGAT level 1 score. The second correlation has also been observed by Salmon et al. [39].

In retrospect, it was noted earlier in this paper that the measurement scale for performance might be too coarse. This fact, coupled with the relatively small sample size, could explain why the correlations, especially with regard to performance, fluctuate so much between the two interfaces.

This research also examined the differences in the mean values for the two scenarios. However, no statistically significant differences were observed. Most notably, the participants had better SAGAT level 3 scores in the first scenario ($p = 0.077$), which could be because they had to be more focused during an unpredictable disaster situation.

6. Conclusions

This paper has described a set of requirements, framework and test implementation of a monitoring system for national-scale critical infrastructure. The brokered agent based architecture is scalable and efficiently integrates diverse critical infrastructure systems. Moreover, the critical infrastructure monitoring system is implemented using open source technology.

The situational awareness oriented design process was used to identify the situational awareness requirements, which were translated into a system design that provides high levels of situational awareness. Novel modeling and analysis methods targeted for large-scale critical infrastructure systems were developed to satisfy the situational awareness requirements. The approach eliminates the need to collect minute details about every system and sub-system and is, therefore, capable of handling critical infrastructure interdependency networks spanning tens of thousands of nodes. The implemented system also estimates the impacts of events and provides estimates of future status.

An important component of the paper is the final situational awareness oriented design step that involved objective (SAGAT) and subjective (SART) measurements of situational awareness performed during "human-in-the-loop" simulations. The evaluations also incorporated SUS in the test battery to evaluate system usability. The measurement methods view the situational awareness construct differently and essentially measured different elements of participant awareness during the evaluations. SAGAT, which is a probe-recall approach, measured the extent to which participants were aware of pre-defined elements in the environment, their understanding of the properties of the elements in relation to the tasks being performed and the potential future states of the elements. SART measured participant self-awareness during the task performance based on understanding, supply and demand ratings without reference to the different elements in the environment. Mixed results were obtained in the evaluations, but it is important to note that good SUS results do not necessarily imply good situational awareness. The findings from this study – as well as previous research – suggest that situational awareness cannot be fully measured by a single metric. Therefore, it is appropriate to employ a battery of different, but compatible, measures.

Acknowledgment

This research was partially supported by the Finnish Funding Agency for Technology and Innovation under TEKES Grant no. 40345/12.

REFERENCES

[1] C. Alcaraz and J. Lopez, Wide-area situational awareness for critical infrastructure protection, *IEEE Computer* vol. 46(4), pp. 30–37, 2013.
[2] A. Altwood, M. Merabti, P. Fergus and O. Abuelmaatti, SCCIR: Smart Cities Critical Infrastructure Response Framework, *Proceedings of the Fourth International Conference on Developments in E-Systems Engineering*, pp. 460–464, 2011.
[3] Apache Software Foundation, Apache ActiveMQ, Wakefield, Massachusetts (http://activemq.apache.org), 2011.
[4] Apache Software Foundation, Apache Camel, Wakefield, Massachusetts (http://camel.apache.org), 2015.
[5] E. Bagheri and A. Ghorbani, A service oriented approach to critical infrastructure modeling, *Proceedings of the Workshop on Service Oriented Techniques*, 2006.
[6] E. Bagheri and A. Ghorbani, UML-CI: A reference model for profiling critical infrastructure systems, *Information Systems Frontiers*, vol. 12(2), pp. 115–139, 2010.
[7] C. Bolstad, A. Costello and M. Endsley, Bad situation awareness designs: What went wrong and why, presented at the *Sixteenth World Congress of the International Ergonomics Association*, 2006.
[8] J. Brooke, SUS – A quick and dirty usability scale, in *Usability Evaluation in Industry*, P. Jordan, B. Thomas, B. Weerdmeester and A. McClelland (Eds.), Taylor and Francis, London, United Kingdom, pp. 4–7, 1996.
[9] E. Casalicchio, E. Galli and S. Tucci, Federated agent-based modeling and simulation approach to study interdependencies in IT critical infrastructures, *Proceedings of the Eleventh IEEE International Symposium on Distributed Simulation and Real-Time Applications*, pp. 182–189, 2007.
[10] E. Castorini, P. Palazzari, A. Tofani and P. Sevillo, Ontological framework for modeling critical infrastructures and their interdependencies, *Proceedings of the Complexity in Engineering Conference*, pp. 91–93, 2010.
[11] M. Endsley, Situation Awareness Global Assessment Technique (SAGAT), *Proceedings of the IEEE National Aerospace and Electronics Conference*, vol. 3, pp. 789–795, 1988.
[12] M. Endsley, *Designing for Situation Awareness: An Approach to User-Centered Design*, CRC Press, Boca Raton, Florida, 2011.
[13] M. Endsley, Situation awareness misconceptions and misunderstandings, *Journal of Cognitive Engineering and Decision Making*, vol. 9(1), pp. 4–32, 2015.
[14] M. Endsley, C. Bolstad, D. Jones and J. Riley, Situation awareness oriented design: From user's cognitive requirements to creating effective supporting technologies, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 47(3), pp. 268–272, 2003.
[15] M. Endsley and E. Connors, Situation awareness: State of the art, *Proceedings of the IEEE Power and Energy Society General Meeting – Conversion and Delivery of Electrical Energy in the 21st Century*, 2008.

[16] M. Endsley, S. Selcon, D. Hardiman and D. Croft, A comparative analysis of SAGAT and SART for evaluations of situation awareness, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 42(1), pp. 82–86, 1998.
[17] M. Endsley, R. Sollenberger and E. Stein, Situation awareness: A comparison of measures, *Proceedings of the Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium Conference*, pp. 15–19, 2000.
[18] T. Erl, *Service-Oriented Architecture: Concepts, Technology and Design*, Prentice Hall, Boston, Massachusetts, 2005.
[19] F. Flammini, N. Mazzocca, A. Pappalardo, C. Pragliola and V. Vittorini, Improving the dependability of distributed surveillance systems using diverse redundant detectors, in *Dependability Problems of Complex Information Systems*, W. Zamojski and J. Sugier (Eds.), Springer, Cham, Switzerland, pp. 35–53, 2015.
[20] N. Giacobe, Application of the JDL data fusion process model for cyber security, *Proceedings of SPIE*, vol. 7710, pp. 77100R-1–77100R-10, 2010.
[21] D. Jones and M. Endsley, Can real-time probes provide a valid measure of situation awareness, *Proceedings of the Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium Conference*, pp. 472–492, 2000.
[22] M. Klemetti, S. Puuska and J. Vankka, Entropy measures in critical infrastructure graphs, *Proceedings of the Seventh Conference of the International Society of Military Sciences*, 2015.
[23] M. Klemetti, S. Puuska and J. Vankka, Entropy as a metric in critical infrastructure situational awareness, *Proceedings of SPIE*, vol. 9825, pp. 98250K-1–98250K-8, 2016.
[24] J. Kopylec, A. D'Amico and J. Goodall, Visualizing cascading failures in critical cyber infrastructures, in *Critical Infrastructure Protection*, E. Goetz and S. Shenoi (Eds.), Springer, Boston, Massachusetts, pp. 351–364, 2007.
[25] A. Koskinen-Kannisto, Situational Awareness Concept in a Multinational Collaboration Environment: Challenges in the Information Sharing Framework, Doctoral Dissertation, Department of Military Technology, National Defense University, Helsinki, Finland, 2013.
[26] L. Lääperi and J. Vankka, Architecture of a system providing a common operating picture of critical infrastructure, *Proceedings of the IEEE International Symposium on Technologies for Homeland Security*, 2015.
[27] T. Lewis, *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation*, John Wiley and Sons, Hoboken, New Jersey, 2006.
[28] Z. Liu and B. Xi, Copula model design and analysis of critical infrastructure interdependencies, *Proceedings of the Nineteenth Annual International Conference on Management Science and Engineering*, pp. 1890–1898, 2012.
[29] S. Marrone, R. Nardone, A. Tedesco, P. D'Amore, V. Vittorini, R. Setola, F. De Cillis and N. Mazzocca, Vulnerability modeling and analysis for critical infrastructure protection applications, *International Journal of Critical Infrastructure Protection*, vol. 6(3-4), pp. 217–227, 2013.
[30] M. Nieminen and M. Koivunen, Visual walkthrough, *People and Computers X: Adjunct Proceedings of the Sixth International Conference on Human-Computer Interaction*, pp. 86–89, 1995.
[31] M. Ouyang, Review of modeling and simulation of interdependent critical infrastructure systems, *Reliability Engineering and System Safety*, vol. 121, pp. 43–60, 2014.
[32] Pivotal Software, Spring Framework, San Francisco, California (http://projects.spring.io/spring-framework), 2015.
[33] S. Puuska, Modeling and Analysis of Critical Infrastructure for Situational Awareness Applications, Master's Thesis, Department of Computer Science, University of Helsinki, Helsinki, Finland, 2016.
[34] S. Puuska, K. Kansanen, L. Rummukainen and J. Vankka, Modeling and real-time analysis of critical infrastructure using discrete event systems on graphs, *Proceedings of the IEEE International Symposium on Technologies for Homeland Security*, 2015.
[35] S. Riihiaho, User testing when test tasks are not appropriate, *Proceedings of the European Conference on Cognitive Ergonomics: Designing Beyond the Product — Understanding Activity and User Experience in Ubiquitous Environments*, article no. 21, 2009.
[36] S. Rinaldi, J. Peerenboom and T. Kelly, Identifying, understanding and analyzing critical infrastructure interdependencies, *IEEE Control Systems*, vol. 21(6), pp. 11–25, 2001.
[37] L. Rummukainen, L. Oksama, J. Timonen and J. Vankka, Visualizing the common operating picture of a critical infrastructure, *Proceedings of SPIE*, vol. 9122, pp. 912208-1–912208-15, 2014.
[38] L. Rummukainen, L. Oksama, J. Timonen and J. Vankka, Situation awareness requirements for a critical infrastructure monitoring operator, *Proceedings of the IEEE International Symposium on Technologies for Homeland Security*, 2015.
[39] P. Salmon, N. Stanton, G. Walker, D. Jenkins, D. Ladva, L. Rafferty and M. Young, Measuring situation awareness in complex systems: Comparison of measures study, *International Journal of Industrial Ergonomics*, vol. 39(3), pp. 490–500, 2009.
[40] J. Sauro, Measuring Usability with the System Usability Scale (SUS), MeasuringU, Denver, Colorado (http://www.measuringu.com/sus.php), 2015.
[41] A. Steinberg, C. Bowman and F. White, Revisions to the JDL data fusion model, *Proceedings of SPIE*, vol. 3719, pp. 430–441, 1999.
[42] R. Taylor, Situational Awareness Rating Technique (SART): The development of a tool for aircrew systems design, *Proceedings of the AGARD Conference on Situational Awareness in Aerospace Operations*, no. 478, pp. 3-1–3-17, 1990.
[43] J. Timonen, L. Lääperi, L. Rummukainen, S. Puuska and J. Vankka, Situational awareness and information collection from critical infrastructure, *Proceedings of the Sixth International Conference on Cyber Conflict*, pp. 157–173, 2014.
[44] W. Tolone, D. Wilson, A. Raja, W. Xiang, H. Hao, S. Phelps and E. Johnson, Critical infrastructure integration modeling and simulation, *Proceedings of the International Conference on Intelligence and Security Informatics*, pp. 214–225, 2004.
[45] United States Computer Emergency Readiness Team (US-CERT), Federal Incident Reporting Guidelines, Washington, DC (http://www.us-cert.gov/government-users/reporting-requirements), 2016.
[46] A. van den Beukel and M. van der Voort, The influence of time-criticality on situation awareness when retrieving human control after automated driving, *Proceedings of the Sixteenth IEEE International Conference on Intelligent Transportation Systems*, pp. 2000–2005, 2013.
[47] C. Wang, L. Fang and Y. Dai, National critical infrastructure modeling and analysis based on complex system theory, *Proceedings of the First International Conference on Instrumentation, Measurement, Computer Communications and Control*, pp. 832–836, 2011.
[48] M. Wang Baldonado, A. Woodruff and A. Kuchinsky, Guidelines for using multiple views in information visualization, *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 110–119, 2000.
[49] S. Yau, D. Huang, H. Gong and H. Davulcu, Situation-awareness for adaptive coordination in service-based systems, *Proceedings of the Twenty-Ninth Annual International Computer Software and Applications Conference*, vol. 2, pp. 107–112, 2005.

[50] S. Yau and J. Liu, Incorporating situation awareness in service specifications, *Proceedings of the Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, 2006.

[51] R. Zimmerman, Decision-making and the vulnerability of interdependent critical infrastructure, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 4059–4063, 2005.

[52] R. Zimmerman and C. Restrepo, Analyzing cascading effects within infrastructure sectors for consequence reduction, *Proceedings of the IEEE Conference on Technologies for Homeland Security*, pp. 165–170, 2009.

P4

BLUE TEAM COMMUNICATION AND REPORTING FOR ENHANCING SITUATIONAL AWARENESS FROM WHITE TEAM PERSPECTIVE IN CYBER SECURITY EXERCISES

by

T. Kokkonen & S. Puuska 2018

In: Galinina O., Andreev S., Balandin S., Koucheryavy Y. (eds) Internet of Things, Smart Spaces, and Next Generation Networks and Systems. NEW2AN 2018, ruSMART 2018. Lecture Notes in Computer Science, vol 11118 pp. 277–288. Springer, Cham

https://doi.org/10.1007/978-3-030-01168-0_26

Blue Team Communication and Reporting for Enhancing Situational Awareness from White Team Perspective in Cyber Security Exercises

Tero Kokkonen and Samir Puuska

Institute of Information Technology, JAMK University of Applied Sciences,
Jyväskylä, Finland
{tero.kokkonen, samir.puuska}@jamk.fi

Abstract. Cyber security exercises allow individuals and organisations to train and test their skills in complex cyber attack situations. In order to effectively organise and conduct such exercise, the exercise control team must have accurate situational awareness of the exercise teams. In this paper, the communication patterns collected during a large-scale cyber exercise, and their possible use in improving Situational awareness of exercise control team were analysed. Communication patterns were analysed using graph visualisation and time-series based methods. In addition, suitability of a new reporting tool was analysed. The reporting tool was developed for improving situational awareness and exercise control flow. The tool was used for real-time reporting and communication in various exercise related tasks. Based on the results, it can be stated that the communication patterns can be effectively used to infer performance of exercise teams and improve situational awareness of exercise control team in a complex large-scale cyber security exercise. In addition, the developed model and state-of-the-art reporting tool enable real-time analysis for achieving a better situational awareness for the exercise control of the cyber security exercise.

Keywords: Cyber Security · Exercise · Training · Situational Awareness · Communication.

1 Introduction

Cyber security is an ongoing process where both organisations and individuals are training, working, and learning continually. Cyber security exercises are an excellent way to train and simultaneously test an organisation's or individual's capabilities under stressful cyber-attack situations. The exercise can be conducted in both public and private sectors. The cyber security strategy of the European Union notices the importance of national and international cyber security exercises [8]. Finland's security strategy for society states several times the importance of regular exercises for improving the resilience against threats [23], whereas Finland's cyber security strategy states that cyber threats are evolving extremely rapidly, and therefore cyber security exercises should be

conducted regularly for improving preparedness and cyber resilience [22]. Handbook for information technology and cyber security exercises [26] lists following exercise types: unannounced live exercises, initiation exercises, staff exercises, decision exercises, management exercises, cooperation exercises and Red Team - Blue Team exercises. The exercise type indicates the primary function of the exercise.

Cyber security exercises are usually organised using various teams with different tasks or missions. These teams are formed based on exercise type, training goals, and available resources and personnel. Blue Team (BT) is a group of people defending their information technology assets against cyber threats. They also report the observations to (simulated) management, create their own situational awareness and maintain their own security posture under cyber-attack. BT is very often modelled after a real organisation, team, or branch. There can be one or many BTs in the exercise that can represent different aspects of the real world. BTs often aim to role-play their normal organisational practices and procedures. Red Team (RT) is a group of people simulating the threat actors in the exercise by making real cyber-attacks against Blue Teams. White Team (WT) is responsible for controlling the exercise, making observations, collecting the data and handling the situational awareness of the exercise [5, 26, 13, 25].

Sometimes the exercise control team is also called EXCON which has similar functions as WT. In that sense, the situational awareness of the WT is extremely important for controlling the exercise and for making the required decisions during the exercise. The communication patterns of the BTs are an important source for understanding what is happening in the exercises from the BT's perspective, and how they are communicating with the co-operation organisations under cyber-attack.

One of the most classical definitions of situational (or situation) awareness is as follows: *"Situation awareness is the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"* [7]. In this study, the term situational awareness (SA) is used. At the first level of SA there is the perception (observations and sensor information), the second level is the comprehension (understanding the current situation) and the third level is the projection (prediction of future events based on the information of earlier states and decision makers' pre-learned history). It is stated that with erroneous SA even the trained decision makers will make incorrect decisions [7]. In the cyber security the objective of SA is to know what is (and will be) the security level of organisation's assets in the networked systems [9].

Cyber security exercises enable a comprehensive platform for studying situational awareness in cyber security and behaviour or efficiency of individuals and teams under cyber-attack. In the study [6] a methodology is proposed for adjustment of situation awareness measurement experiments within the context of a cyber security exercise. The author of [10] states that cyber security exercises can be used as an empirical study of situation awareness in cyber security. Also, the paper [5] deploys cyber security exercise data for profiling the attacker. Accord-

ing to the authors of the studies [4, 3], training and exercises have an important role for improving the competencies in the defence of the cyber security assets and for achieving the required level of preparedness especially in the resilience of critical infrastructure.

Situational awareness is important for all involved teams in the exercise. However, WT is required to have an understanding of the SA of the BTs in exercise in order to effectively adjust and steer the exercise towards fulfilling the desired learning and testing goals. Traditional monitoring of technical details of the exercise environment supplemented with the analysis of communication patterns provides an extensive view into Blue Team behaviour.

This study presents the study of Blue Team communication patterns and based on that the implementation of the state-of-the–art reporting tool for enhancing the SA of the White Team during the complex and hectic cyber security exercise. First the Finland's national cyber security exercise is introduced, the event timelines are are studied, and analysis is made. In addition, the reporting tool is developed and studied to produce incident reports for enhancing the SA of the White Team. Finally, the conclusions are done, and future research ideas are found and introduced.

2 Finland's National Cyber Security Exercise

Finland's national cyber security exercise has been conducted annually since 2013 and every year, the Cyber Range of Finland's national cyber exercise has been Realistic Global Cyber Environment (RGCE) developed by JAMK University of Applied Sciences Institute of Information Technology [18].

Finland's national cyber security exercise of 2017 was executed from 8th of May to 11th of May and it was commanded by the Ministry of Defence with The Security Committee. The RGCE Cyber Range and the overall implementation was conducted by JAMK University of Applied Sciences. There were more than 100 individuals participating in the exercise forming several co-operating Blue Teams communicating with each other according to their operational tasks. The aim of the exercise was to practice co-operation between security organisations and security network organisations in Finland during cyber-attacks or incidents for verifying the performance of the participant organisations and ensuring their further development [18].

As described in the aim of the exercise, the Blue Teams of the exercise were formed from different security authorities of Finland. All of them were acting, communicating and co-operating according to their real operational tasks during the realistic cyber attacks of several simulated threat actors. Some of the Blue Teams mainly defend their own assets whereas some Blue Teams have highly co-operational role and act and communicate actively in accordance with that role.

2.1 RGCE Cyber Range

RGCE is a fully operational Cyber Range that mimics the structures, services and traffic of the real Internet. It allows the usage of real IP addresses and global GeoIP information with realistic end user traffic patterns automatically generated by botnet based special software. RGCE is a closed environment, which allows usage of real attacks or malware. [14, 12]

3 Event Timelines

Cyber security exercises consist of several components forming the core which the White Team uses to direct the overall flow. A typical exercise contains a background story that sets the general tone and mindset for the trainees. Several threat actors are created to portray real-world counterparts, such as hactivist groups and more advanced organisations. Based on these actors and their modus operandi, various attack scenarios are prepared. The scenarios may include technical exploitations, denial-of-service attacks, social engineering, and advanced directed cyber operations.

3.1 Injects

Injects are pre-prepared actions in the Cyber Range. They are modelled after the threat actor's simulated campaigns. For example, a malicious group may want to use a denial-of-service (DoS) or a distributed denial-of-service (DDoS) attack to mask a more advanced exploit, targeted at one team. This could be achieved by two injects, one for each type of attack. The schedule for injects is drafted at the planning stage. However, due to the live nature of cyber exercises, White Team may choose to adjust their timing, targets or their potential execution, depending on the Blue Team response. Adjusting overlapping incidents and injects to support learning goals and desired stress levels is crucial for a successful exercise.

For the studied exercise, dozens of injects were prepared to simulate the cyber attack campaigns of threat actors. There were several realistic threat actors modelled and simulated in the exercise and the injects were prepared to simulate the behaviour of those threat actors. The attack campaigns varied from volumetric DoS/DDoS campaigns to targeted advanced persistent threat (APT) attacks including for example realistic behaviour of threat actors in social media.

Figure 1 illustrates the duration of the injects during the cyber security exercise. When WT decides to activate an inject, the actual time is recorded, as well as the moment when the inject in question is marked as 'executed', i.e. it does not require any further work from any of the teams. Figure 1 shows, that the approximate workload is relatively evenly distributed inside each exercise day, first and last being less intensive. This was the desired goal in the planning stage.

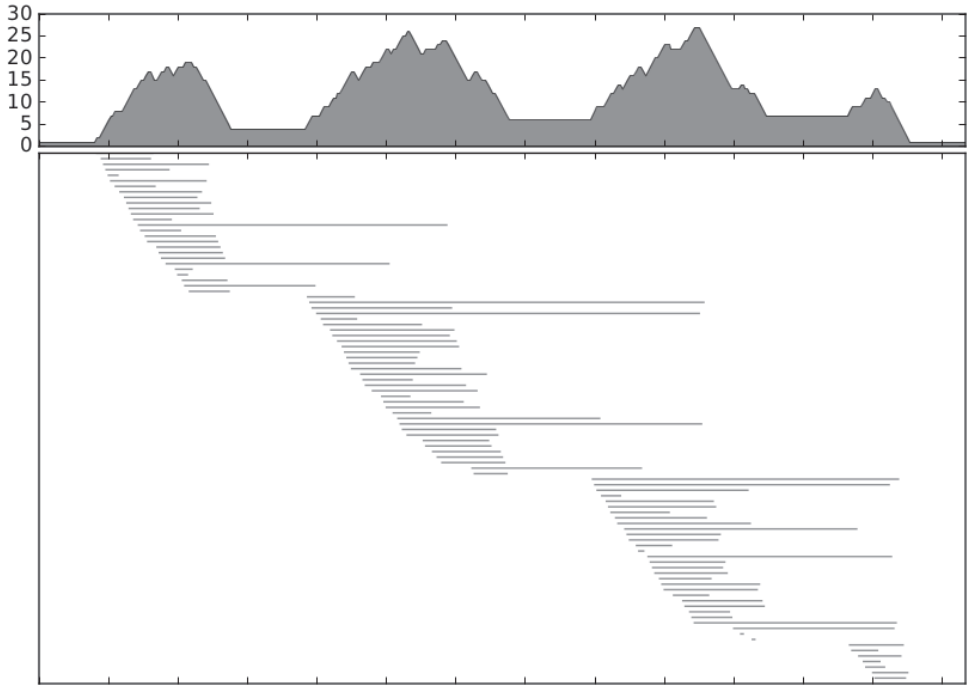


Fig. 1: Inject timing, durations (lower), and cumulative sum (upper) during the Exercise.

3.2 Communication Methods

Blue Teams were given various common methods for communicating between groups and internally. Each team had corporate email-accounts, two kinds of direct messaging options, and VOIP phones. Overall, the teams preferred e-mail over other forms of communication. Therefore, this study focuses on e-mails, and data fusion between other systems is considered as future work.

4 Analysis

Although figure 1 illustrates the approximate amount of desired work, it does not tell how the exercise teams actually react to the injects. In some cases the exercise teams may miss the inject entirely or fail to take appropriate measures. Direct monitoring or questionnaires disturb the flow of the exercise and require extra personnel.

E-mail patterns were analysed to see what communication patterns teams use during incidents. The mail headers were extracted from mail servers and analysed and visualised using Cytoscape software [24].

4.1 Team Communication Patterns

BTs in the exercise played several different roles. For example, one BT formed a common networking and service platform, which includes physical networks, as

well as workstations and intranet services, and another BT was a cyber security service organisation offering services to all other teams.

During the exercise tens of thousands of emails were sent and received, also including an e-mail-based Denial of Service -attack, as well as general spam, and e-mails from automated reporting systems. BTs also forwarded information to each other using large mailing lists. Some teams included their own address into these lists, and therefore received many copies of their own mails. White Team also answered to requests and inquiries that were directed to higher levels of organisations not occupied in the exercise.

Figure 2 illustrates all used message paths between parties. Red nodes represent attacker-controlled domains, coloured ones are the Blue Teams. Edge colour indicates the sending party. The graph shows that Teams two and five never communicated directly, even though they should have.

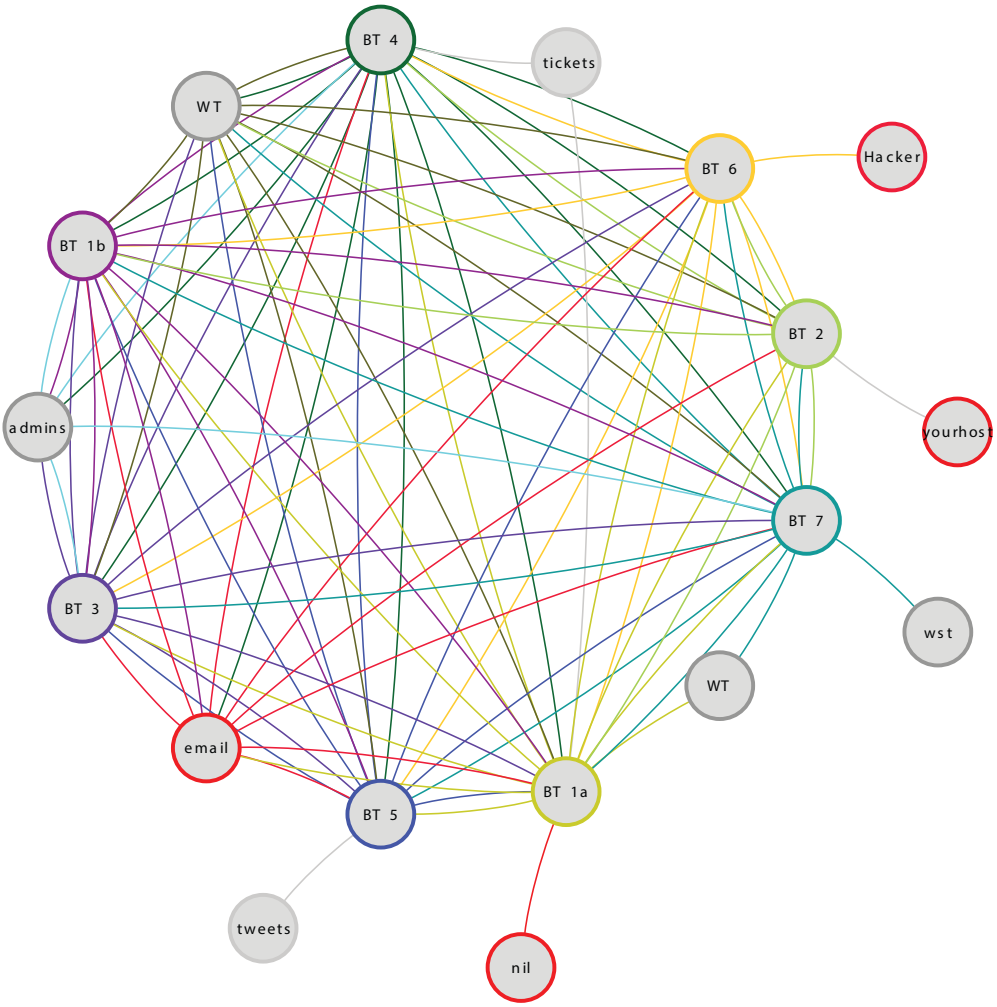
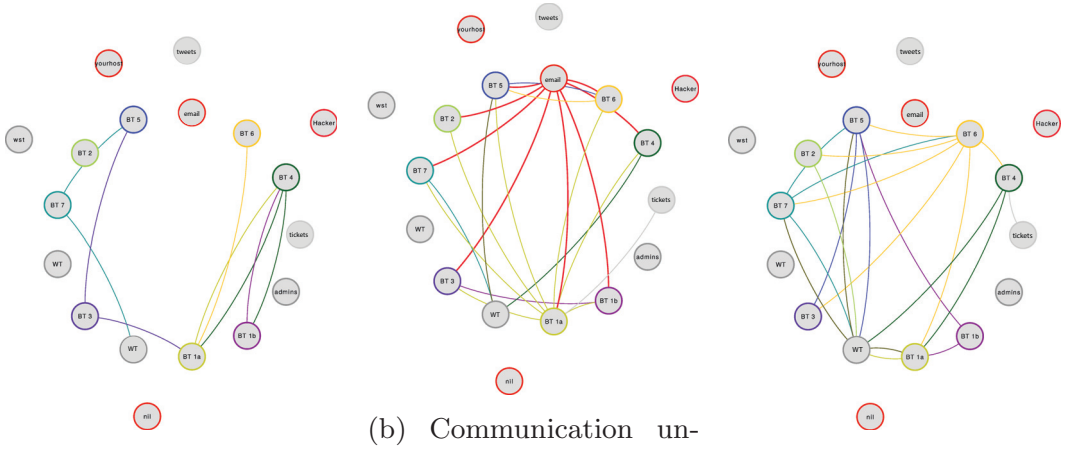


Fig. 2: The complete communication graph between domains.

The mailing patterns mostly reflect the nature and purpose of each team. Blue Team one, which was responsible for the core services, communicated with all other organisations actively. Their mails informed the organisations that were using their services about various disruptions, estimated repair times, and detected threats. Blue Team two was noticeably less active. They sent only a few notices of service disruptions, and mainly co-operated with Blue Team one, even though they were kept up to date by other teams. Blue Team three mostly co-operated with Blue Team five, which was expected. Blue Team six communicated actively with every other team, delivering threat intelligence and analysis services. Blue Teams four and one were also targeted by external Denial of Service and phishing campaigns. This may have affected their capability to send and receive mails.

In figure 3a, a typical set of service requests and responses is made. They indicate that the teams still have control over their infrastructure, and are able to take defensive measures. Figure 3b illustrates a phishing attempt, which later evolved into a spamming attack. Grey nodes represent mailboxes belonging to non-playing teams, while red nodes are controlled by threat actor (RT). In figure 3c Blue Team six has detected an unusually intensive port scanning originating from the Internet. The team informs others, and it can be seen that one team asks for more details.



(a) Generic communication during the exercise.

(b) Communication under spam and phishing attacks.

(c) Communication under port scanning attack.

Fig. 3: Example of communication patterns.

Although the analysis of communication patterns revealed some omissions and errors that teams made, it does not have enough information for White Team to form a robust SA. Also, the analysis of communication pattern is not conducted in the real time and more real time reporting tool is required for improving the situational awareness of White Team. It can be concluded, that a special real time reporting system is required for obtaining data and understanding the Blue Team behaviour during the complex cyber security exercise.

5 Reporting Tool for Improving White Team SA

Situational awareness is required as a basis for decision making. OODA loop (Observation-Orientation-Decision-Action) is a classical model for decision making [21, 15]. Another similar decision-making loop is introduced in four stages of an adaptive security architecture (Predict-Prevent-Detect-Respond) [17]. When reflected to both of those loops and earlier introduced definition of SA, SA is an extremely important element of decision making. When considering different data from different sources or sensors, there is a requirement for data fusion or multi-sensor data fusion, which is a process of synthesising overlapping and scattered data from the different sensors or sources to the user for achieving comprehensive SA of focused events [11, 2].

In the cyber security exercises, the Blue Team reporting tool for gathering the SA is required in two functions. First the Blue Teams report (automatically from sensors or manually) their observations to the tool and forms their SA based on data fusion. Secondly, White Team is able to monitor what the Blue Teams are reporting and what mitigation actions they are executing [16].

The developed Reporting Tools was tested in the cyber security exercise in the industrial domain [20]. Industrial cyber security exercise is piloted in the project of the European Regional Development Fund/Leverage from the EU 2014-2020, called JYVSECTEC Center and managed by JAMK University of Applied Sciences Institute of Information Technology.

5.1 Reporting Process and Software Tool

A specialised reporting process and a supporting state-of-the-art software tool for Blue Teams was developed with the aim that the new system would lower the barrier for reporting. The previous systems failed to encourage the teams or reporting actionable information. Although the teams did use earlier tools to report events, the messages were short, uninformative, and untimely. In addition, the earlier platform was cumbersome, which further discouraged reporting. Reporting is seen in Blue Teams as an unnecessary artificial chore that hinders their ability under the cyber-attacks or incidents.

The goal of development was to construct a reporting tool and process that would be unobtrusive and quick to use. Comprehensive reporting was encouraged by providing a template which contained necessary headings and hints what to put under them. GUI with muted colours was opted to use instead of the console-based solutions.

5.2 Reporting Format

For helping the trainees during the complex exercise scenarios, the reporting format is kept relatively simple; it borrows elements from military-style situation report structure. Table 1 presents the main elements of the format. In addition to the presented elements, each report has a time-stamp and title.

Table 1: Report template fields, translations, and purpose.

Field (in Finnish)	**Field (translated in English)**	**Purpose**
Havainnon laatu	Type of observation	What is being reported? Error condition, support request, malicious program, etc.
Tapahtuma	Incident	What has happened?
Seuraukset	Consequences	What impact will this incident cause? What further measures will be likely taken to mitigate the impact?
Tarkennukset	Further information	Additional details about the incident or of the overall situation.
Paikka	Place	Place, if relevant

A formal language was constructed for describing the reporting format in order to construct domain specific language (DSL). This domain specific language (DSL) allows the reports to be both human and machine readable. DSL is also expandable; other message types can be added in the future. The DSL was also equipped with syntax highlighting in the tool. As the DSL is verified using a formal language parser, the program can also notify user if values are missing or invalid.

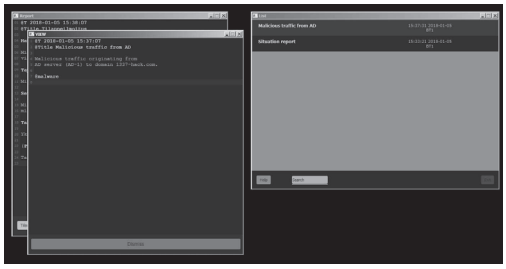
The main view is illustrated in figure 4a By default, the user sees two windows, one of which lists all reports made by his/her team, the other window is for creating a new report. By clicking the reports, they can be opened into a new window and examined separately. The screen-shot shows one additional window that the user has opened.

Figure 4b is a screen-shot of the reporting screen. For keeping the tool simple during the complex and hectic exercise, there are only two buttons and one syntax indicator present in the editor. The button labelled Tilanneilmoitus (Situation report) will fill the editor with the report template. The indicator states if the document does not conform to our DSL specification. The reporting window is a text editor with additional syntax highlighting features.

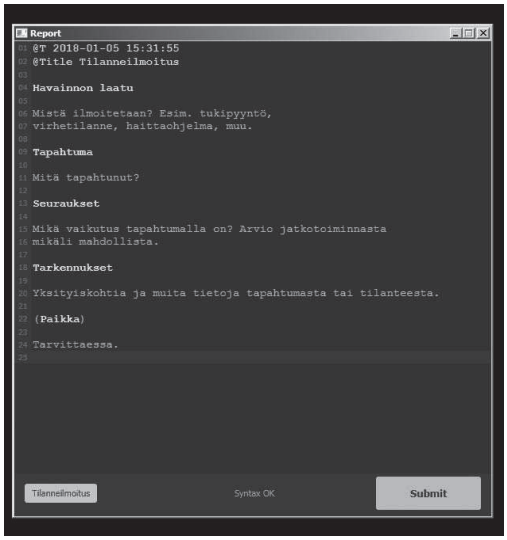
The tool was implemented using Java programming language and JavaFX UI framework, making the tool cross-platform ready [19]. The program utilises a message bus for synchronising messages between team members and delivering a copy of each message to White Team. Our implementation used Apache ActiveMQ message bus for communication [1].

6 Conclusion

Monitoring Blue Team communication provides further insight into both exercise status and team behaviour. As the analysis suggests, communication monitoring can be a useful tool in measuring Blue Team performance during the cyber security exercise. The analysis revealed several omissions made by the Blue Teams.



(a) Main view of the application. The list shows past reports, and the top window shows one of them in full detail. The report editor is in the background.



(b) Report editor with the report template loaded.

Fig. 4: Screen shots of reporting tool.

In addition, although the overall inject timing was successful, some teams might have benefited from intense workload.

When planning the injects, it could be useful to consider which teams are affected, and who is responsible for keeping them informed. By implementing real-time communication monitoring, the White Team can efficiently tell if the teams are acting correctly.

By using e-mail graphs in conjunction with other monitoring mechanisms, real-time mail visualisation aids White Team to build a more robust situational awareness over the exercise. This allows more fine tuned and accurate control, as well as more comprehensive results from the exercise.

However, the special reporting system is required to reliably monitor the Blue Team behaviour in real-time during the cyber security exercise. This requires additional timely reports from the Blue Teams, and a convenient, non-intrusive way for writing and delivering them. A specialised report format and state-of-the-art software tool was developed for achieving this goal. The tool was tested in the cyber security exercises within the industrial domain. It will also be used in the future exercises with improvements suggested in the initial tests.

Future work in the communication monitoring includes automating the message parsing and visualisation process so, that it is readily available to White Team during the exercise. This includes the development of a better visualisation system for monitoring purposes. In the future graphics will be designed to visualise multi-edged graphs efficiently for SA purposes. Future work with the reporting system will be more visualised SA of Blue Team behaviour for certain exercise inject and improvements of BT SA used for BTs' tactical leading and decision making.

Acknowledgment

This research is partially done in JYVSECTEC Center project funded by the Regional Council of Central Finland/Council of Tampere Region and European Regional Development Fund/Leverage from the EU 2014-2020.

References

1. The Apache Software Foundation: Apache activemq. http://activemq.apache.org/, accessed: 23 April 2018
2. Azimirad, E., Haddadnia, J.: The Comprehensive Review On JDL Model In Data Fusion Networks: Techniques and Methods. (IJCSIS) International Journal of Computer Science and Information Security **13**(1), 53–60 (Jan 2015)
3. Brilingaitė, A., Bukauskas, L., Krinickij, V., Kutka, E.: Environment for Cybersecurity Tabletop Exercises. In: Pivec, M., Josef, G. (eds.) ECGBL 2017 11th European Conference on Game-Based Learning, pp. 47–55. Academic Conferences and Publishing Limited (2017)
4. Brilingaitė, A., Bukauskas, L., Kutka, E.: Development of an Educational Platform for Cyber Defense Training. In: Scanlon, M., Nhien-An, L.K. (eds.) Proceedings of the 16th European Conference on Cyber Warfare and Security, pp. 73–81. Academic Conferences and Publishing Limited (2017)
5. Brynielsson, J., Franke, U., Tariq, M.A., Varga, S.: Using Cyber Defense Exercises to Obtain Additional Data for Attacker Profiling. In: 2016 IEEE Conference on Intelligence and Security Informatics (ISI). pp. 37–42 (Sept 2016). https://doi.org/10.1109/ISI.2016.7745440
6. Brynielsson, J., Franke, U., Varga, S.: Cyber situational awareness testing. In: Akhgar, B., Brewster, B. (eds.) Combatting Cybercrime and Cyberterrorism: Challenges, Trends and Priorities, pp. 209–233. Springer International Publishing (2016)
7. Endsley, M.: Toward a Theory of Situation Awareness in Dynamic Systems. Human Factors **37**(1), 32–64 (1995). https://doi.org/10.1518/001872095779049543
8. European Comission: Cybersecurity Strategy of the European Union: An Open, Safe and Secure Cyberspace (Feb 2013)
9. Evesti, A., Kanstrén, T., Frantti, T.: Cybersecurity Situational Awareness Taxonomy. In: 2017 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA). pp. 1–8 (June 2017). https://doi.org/10.1109/CyberSA.2017.8073386
10. Franke, U., Brynielsson, J.: Cyber situational awareness – A systematic review of the literature. Computers & Security **46**, 18–31 (oct 2014)
11. Han, X., Sheng, H.: A New Method of Multi-Sensor Data Fusion. In: 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC). pp. 877–882 (Oct 2017). https://doi.org/10.1109/ITOEC.2017.8122479
12. JAMK University of Applied Sciences, Institute of Information Technology, JYVSECTEC: Rgce cyber range. http://www.jyvsectec.fi/en/rgce/, accessed: 23 April 2018
13. Kick, J.: Cyber exercise playbook. The MITRE Corporation https://www.mitre.org/sites/default/files/publications/pr_14-3929-cyber-exercise-playbook.pdf (2014), accessed: 23 April 2018

14. Kokkonen, T., Hämäläinen, T., Silokunnas, M., Siltanen, J., Zolotukhin, M., Neijonen, M.: Analysis of Approaches to Internet Traffic Generation for Cyber Security Research and Exercise. In: Balandin, S., Andreev, S., Koucheryavy, Y. (eds.) Lecture Notes in Computer Science, pp. 254–267. Springer International Publishing (2015)
15. Lenders, V., Tanner, A., Blarer, A.: Gaining an Edge in Cyberspace with Advanced Situational Awareness. IEEE Security Privacy **13**(2), 65–74 (Mar 2015). https://doi.org/10.1109/MSP.2015.30
16. Lötjönen, J.: Requirement specification for cyber security situational awareness, Defender's approach in cyber security exercises. Master's thesis, JAMK University of Applied Sciences (Dec 2017)
17. van der Meulen, R.: Build Adaptive Security Architecture Into Your Organization. `https://www.gartner.com/smarterwithgartner/build-adaptive-security-architecture-into-your-organization/` (Jun 2017), accessed: 23 April 2018
18. Ministry of Defence Finland: The authorities of the state administration are trained in cyber-skills in Jyväskylä - Valtionhallinnon viranomaiset harjoittelevat kyberosaamista Jyväskylässä 8.-11.5.2017, official bulletin 3th of may 2017. `https://www.defmin.fi/ajankohtaista/tiedotteet/valtionhallinnon_viranomaiset_harjoittelevat_kyberosaamista_jyvaskylassa_8.-11.5.2017.8418.news` (May 2017), accessed: 23 April 2018
19. Oracle Corporation: Java programming language. `http://www.oracle.com/technetwork/java/index.html`, accessed: 23 April 2018
20. Pajunen, D.: Cyber security is ensured with genuine exercises. `https://www.fingridlehti.fi/en/cyber-security-ensured-genuine-exercises/` (Sep 2017), accessed: 23 April 2018
21. Révay, M., Líška, M.: Ooda loop in command control systems. In: 2017 Communication and Information Technologies (KIT). pp. 1–4 (Oct 2017). https://doi.org/10.23919/KIT.2017.8109463
22. Secretariat of the Security Committee: Finland's Cyber security Strategy, Government Resolution 24.1.2013 (Jan 2013)
23. The Security Committee: Security Strategy for Society, Government Resolution 2.11.2017 (Nov 2017)
24. Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research **13**(11), 2498–2504 (2003). https://doi.org/10.1101/gr.1239303
25. Sommestad, T., Hallberg, J.: Cyber Security Exercises and Competitions as a Platform for Cyber Security Experiments. In: Jøsang, A., Carlsson, B. (eds.) Secure IT Systems: 17th Nordic Conference, NordSec 2012, Karlskrona, Sweden, October 31 – November 2, 2012. Proceedings. pp. 47–60. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
26. Wilhelmson, N., Svensson, T.: Handbook for planning, running and evaluating information technology and cyber security exercises. The Swedish National Defence College, Center for Asymmetric Threats Studies (CATS) (2014)

P5

ANOMALY-BASED NETWORK INTRUSION DETECTION USING WAVELETS AND ADVERSARIAL AUTOENCODERS

by

S. Puuska, T. Kokkonen, J. Alatalo & E. Heilimo 2019

In: Lanet JL., Toma C. (eds) Innovative Security Solutions for Information Technology and Communications. SECITC 2018. Lecture Notes in Computer Science, vol 11359 pp. 234–246. Springer, Cham.

https://doi.org/10.1007/978-3-030-12942-2_18

Anomaly-based Network Intrusion Detection using Wavelets and Adversarial Autoencoders

Samir Puuska, Tero Kokkonen, Janne Alatalo, and Eppu Heilimo

Institute of Information Technology, JAMK University of Applied Sciences, Jyväskylä, Finland
{samir.puuska, tero.kokkonen, janne.alatalo, eppu.heilimo}@jamk.fi

Abstract. The number of intrusions and attacks against data networks and networked systems increases constantly, while encryption has made it more difficult to inspect network traffic and classify it as malicious. In this paper, an anomaly-based intrusion detection system using Haar wavelet transforms in combination with an adversarial autoencoder was developed for detecting malicious TLS-encrypted Internet traffic. Data containing legitimate, as well as advanced malicious traffic was collected from a large-scale cyber exercise and used in the analysis. Based on the findings and domain expertise, a set of features for distinguishing modern malware from packet timing analysis were chosen and evaluated. Performance of the adversarial autoencoder was compared with a traditional autoencoder. The results indicate that the adversarial model performs better than the traditional autoencoder. In addition, a machine learning pipeline capable of analyzing traffic in near real time was developed for data analysis.

Keywords: Adversarial Autoencoder · Intrusion Detection · Anomaly Detection · Haar Wavelets.

1 Introduction

The Internet is becoming more secure as encryption becomes more ubiquitous and new standards are adopted. Web pages and other related assets that make up modern web applications are more often transferred using Transport Layer Security (TLS). In addition to providing security to end users, it also allows malicious actors to leverage encryption for evading detection. Therefore, it is extremely important to know the situation of your own valuable assets in the network. For accomplishing that task, one must maintain good visibility into the network, despite of increasing encryption.

Artificial intelligence (AI) and its applications for cyber security are active and growing research fields. Pham et al. compared various machine learning techniques commonly used for intrusion detection [20], while Dhingra et al. outlined several different application areas and challenges for AI in the enterprise information security landscape [6]. Hendler et al. used neural networks for detecting malicious PowerShell commands [9]. Various novel approaches, such as neural immune detectors, Short-Term Memory Recurrent Neural Networks, and Stacked

Auto-Encoders (SAE) have also been studied for attack detection [13, 14, 25]. Intrusion detection systems (IDS) can be divided into anomaly-based detection (anomaly detection) and signature-based detection (misuse detection). Anomaly detection has capability to detect unknown attack patterns; however, anomaly detection usually generates a large amount of false positive indications [19].

Modern intrusions and malware are tasked from the Internet by malicious actors using so-called command and control (C2) channels. Modern malware utilizes various techniques, such as encryption and steganography, for avoiding the detection of communication with a C2 server. TLS is an extremely good way of hiding the command and control traffic because it has become almost ubiquitous, and the recent efforts at hardening TLS infrastructure, such as certificate preloading or easily obtainable free legitimate certificates, have made certificate bumping and other deep inspection methods unreliable. This paper focuses on malware that utilizes TLS for evasion.

This paper presents an anomaly detection -based IDS that leverage Haar wavelet transforms and Adversarial Autoencoders (AA). First, the reasoning about selected features and methods is presented. Next, the implemented solution and validation results are shown. Finally, future research topics are given along with a discussion.

2 Feature Engineering and Selection

Feature selection is the key element in anomaly detection, determining the maximal effectiveness of the detection capability. Chandola et al. [4], and Sommer et al. [22] both listed several challenges in applying machine learning to anomaly detection. One of the challenges they mentioned is how to select features that actually vary between legitimate and malicious traffic, and defining an effective boundary between them. They also noted that when a malicious actor is involved, the adversary is able to adapt.

Due to the increasing ratio of encrypted traffic, the features cannot utilize the payload of the network packets. In the Internet Protocol (IP) packets, the fields cannot be encrypted and are available for feature engineering. Packet timings, and TLS connection parameters, such as handshake parameter negotiation, are also available.

Networks and Internet traffic are not static in volume or content. They experience considerable variance depending on many factors such as workday cycles, scheduled software updates, or changes in workforce structure. More formally, time series based on our features are non-stationary.

The aim of feature selection is to use as much feature engineering as possible to filter out variances in data that are known to be irrelevant. In addition, the features used must not be readily attacker controlled or circumvented.

2.1 TLS Fingerprints

In the TLS handshake the client and server agree what cryptographic suites they will use. The client sends its preferred suites in preferred order in a package

dubbed "ClientHello" [21]. The order, number, and types of these suites vary considerably between web browsers, desktop applications and other programs, thus forming a sort of fingerprint.

Common malware and various APT-simulation tools, such as CobaltStrike[1], Empire[2], and Meterpreter[3], were analysed. It was discovered that their TLS handshakes were either unique or different than legitimate programs. Specific versions of Firefox browsers used in Kali Linux[4], a well known penetration testing tool collection, was also detected. This is in line with previous research; for example, Husák et al. obtained similar results when fingerprinting applications [10]. Although preliminary look into this feature gave extremely positive results, it is also something that the adversary may choose to change, as e.g. TOR meek[5] does. The feature is still useful in a limited manner, it groups applications reliably and only the most sophisticated adversary can tailor the malware traffic to look like the one a particular target organization is using. It should be noted that e.g. PowerShell environment, often used for running malicious code, does not allow the scripts to select preferred suites. We did not use TLS fingerprints in the neural network input data.

The malware used in this research was tasked to beacon to the C2 server from several hours to days. After that the malware was used to perform various malicious activities, such as listing processes, transferring files, taking screenshots, and for further lateral movement on the internal networks. First infection was achieved using either phishing e-mails, or custom-made zero-day exploits.

2.2 Network Flows and Time Series

Network flows form a natural time series, especially when considering those made with Transmission Control Protocol (TCP). Depending on the application, these flows will vary with respect to duration, number of packets transmitted and received, and periodicity, among other characterizing statistics. As previously stated, these time series are non-stationary. This is partly due to the inherent nature of TCP flows, as well as the aim to keep once-negotiated tunnels up for subsequent data transfer, rather than renegotiate. This is also true for TLS, which in virtually any application, runs on top of TCP.

There are several well-know legitimate use cases for TLS encrypted connections. The most ubiquitous one is the World Wide Web; virtually every major web application is accessible or mandates the use of TLS. Virtual Private Networks (VPN) may also be deployed on top of TLS. Compared to the web browsing, these connections can be longer-lived, and their activity is more varying. The third major category is desktop applications, which use TLS to connect securely to their back-ends which may reside either on an internal network or the Internet.

[1] https://www.cobaltstrike.com/
[2] https://www.powershellempire.com/
[3] https://www.offensive-security.com/metasploit-unleashed/about-meterpreter/
[4] https://www.kali.org/
[5] https://trac.torproject.org/projects/tor/wiki/doc/meek

As for malware, TLS provides a practical channel for communicating with Command and Control (C2) servers. It blends in easily with legitimate network traffic, and in many deployments is permitted through firewalls. However, these connections are not usually similar to the legitimate use cases. In the analyzed malware and APT tools, the connections were very short-lived when there were no instructions available for them in their C2 server. When they are tasked to e.g. transfer files or take screenshots, the connections looked different. Based on these observations it was concluded that an aggregation of TLS connections using the IP address or Server Name Indication (SNI) record, the result will form a descriptive time series usable for anomaly detection. This series can be constructed from packet timings and sizes made into an impulse signal, where received packets have negative values, sent packets positive values and where the impulse values are the packet sizes.

2.3 Analysis using Haar Wavelets

There are many options for characterizing a time series. It is important to use a representation that retains the essential features for the classification task at hand without overfitting the data. By considering overfitting also at this stage, the input to the classification algorithm can be made less noisy. Due to the non-stationary nature of our data, the methods at our disposal are somewhat limited. The differing lengths of the series also needs to be addressed.

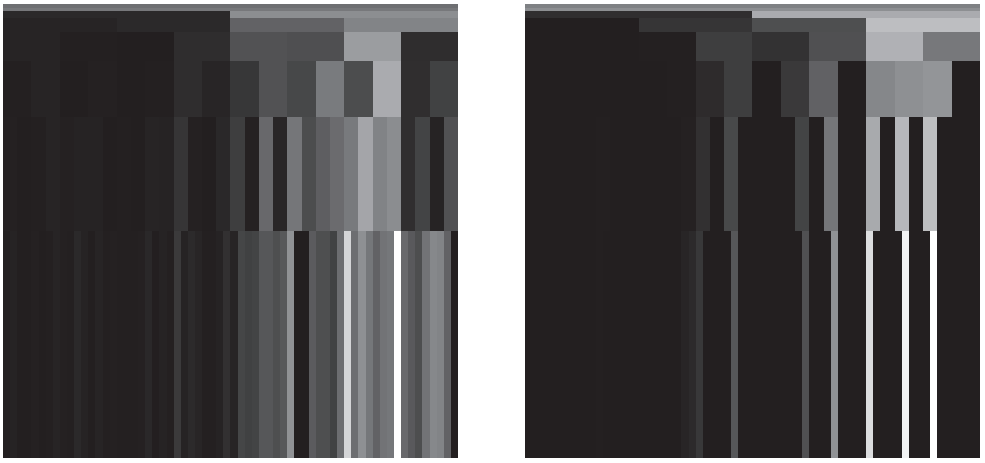
There are two main categories for mathematical time series representations, data-adaptive and non data-adaptive [15]. Haar wavelets [8], a non data-adaptive representation, was chosen. The transform contains both time and frequency elements, and is therefore advantageous for data which is both non-stationary and sparse [5, 3].

Figure 1 illustrates the result of decomposition as it is used in this study. The image represents eight of the lowest frequency coefficient layers from the wavelet transformation result. Brighter areas represent higher coefficient values and the black areas have coefficients values very close to zero. The number of coefficient samples doubles on each layer when more layers are taken. The new layers represent higher and higher frequencies. The solution is designed to examine the long, low frequency traffic patterns so it is safe to discard the high frequency layers. In this study, only the eight coefficient layers that represent the lowest frequencies are kept.

2.4 Adversarial Autoencoders

The wavelet transform provides a starting point for anomaly detection. However, the comparison of the decomposition results is a non-trivial task. There are no obvious ways to assign probabilities to real-valued time series or their transformations in this dataset.

Autoencoders are constructed using artificial neural networks that attempt to reconstruct their input using a relatively small hidden (latent) layer, in a fashion similar to the Principal Component Analysis. Adversarial autoencoders



(a) Legitimate web browser traffic to an authentication portal.

(b) Malicious traffic caused by a CobaltStrike beacon.

Fig. 1: Wavelet decomposition, illustrated here with scalograms, for both legitimate and malicious TLS traffic samples.

have several desired properties for anomaly detection, especially in our use case. Adversarial autoencoders (AA) combine ideas from traditional autoencoders and generative adversarial networks, turning autoencoders into generative models. AAs are suitable for unsupervised learning, or they can also be used in supervised or semi-supervised fashion [17, 7]. Their advantage over traditional AAs is the ability to influence what distribution the hidden layer should approximate. This allows the model to learn additional variance that is not present in the training data, making it less likely overfitted. Adversarial autoencoders use generative adversarial neural networks for regulating the distribution of the autoencoder's latent space (the latent). The encoder of the network is trained to fool the discriminator by generating vectors similar to the chosen distribution, while the discriminator is trained to determine if the sample is generated or from the chosen distribution. Meanwhile, the decoder is trained to reconstruct the input data from the latent space. [17] The reconstruction loss and generation loss are optimized for each batch using separate optimizers, hence the calculated gradients are applied in turns.

Figure 2 shows the general architecture of the neural network design developed during this research. It is based on the architecture proposed by Makhzani et al. [17]. The AA variant in this study (TLS-AAE) is trained unsupervised, and regularized with a continuous distribution. This allows the TLS-AA to better reconstruct input variants not present in the training dataset, resulting in a lower reconstruction error than a traditional autoencoder. Although the reconstruction of the new variants is beneficial for reducing the number of false positives, the network might learn to reconstruct anomalies as well. To counter the unwanted variants, a one-hot categorical distribution was imposed into softmax of the latent, dividing the latent space into clusters of continuous distributions. The amount of clusters can be set using a parameter, that is 20 in this study.

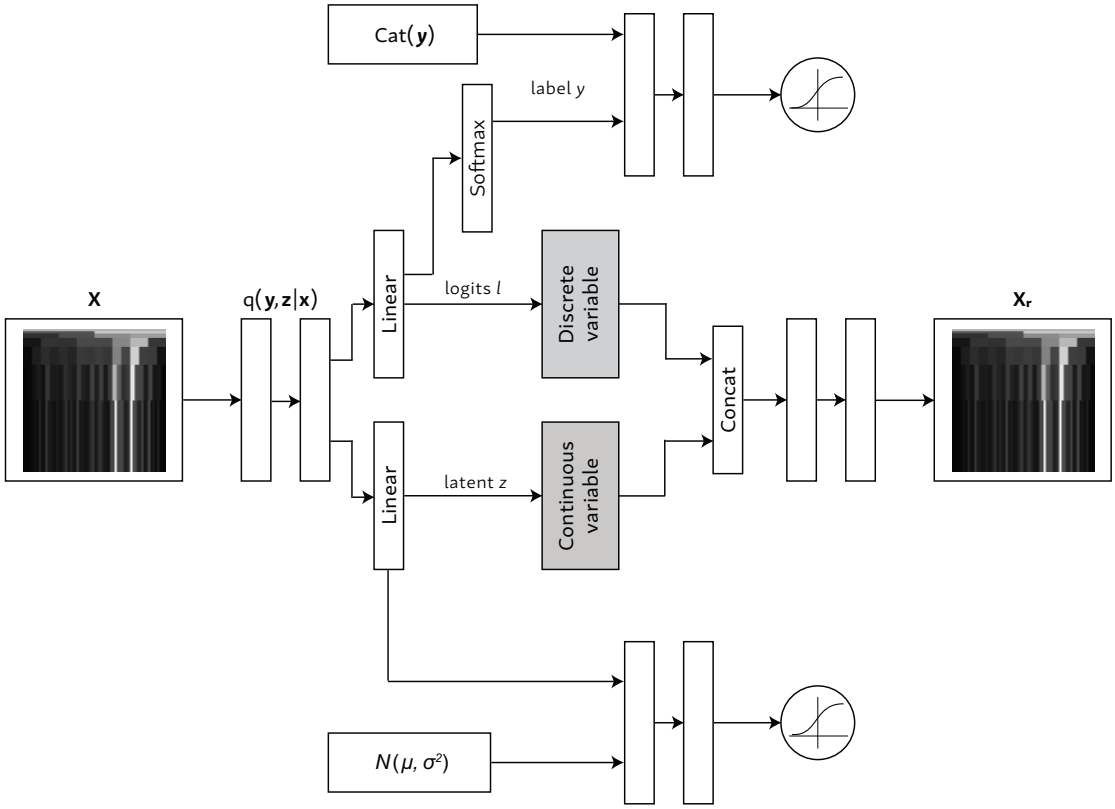


Fig. 2: The general architecture of the neural network design used in this research. The encoder's outputs latent z and label y are enforced to match the selected distributions by establishing a two-player adversarial min-max game between the discriminator and the generator/encoder. [17]. The circles represent adversarial loss, while X_r represents the reconstruction.

The number of clusters does not determine or depend on the labels or classes the network can predict. The TLS-AAE, trained in an unsupervised fashion, uses the classes as a way to utilize the latent space more efficiently.

By utilizing the discrete variable y, the TLS-AAE can output what it considers the best cluster for each input. The clusters are used in a cost function; the cost function is added to the cluster variable (logits l), and it penalizes small Euclidean distances between any two points belonging to different clusters. If the distance between cluster boundaries is over a chosen threshold, it is set to zero. Since the TLS-AAE uses a Gaussian distribution as the continuous distribution, the cluster threshold is set to the length of three standard deviations to ensure that the probability of outlier variants between clusters is minimized. Equation 1 is the cost function, where C is a set containing point sets for each cluster, and $d_{\texttt{min}}$ is the desired minimum distance between any two clusters.

$$L(C, d_{\texttt{min}}) = \frac{1}{|C|} \sum_{x \in C_i} \sum_{y \in C_j,\ i \neq j} \max\{0, d_{\texttt{min}} - ||x - y||^2\} \tag{1}$$

By utilizing several different optimization targets the latent space can be constructed to approximate what the authors believe are reasonable assumptions about the nature of TLS connections. This reasoning stems from the idea that the TLS traffic can be divided into several categories depending on the application responsible for generating it; web browsing, music streaming services, and malware should form distinct clusters. The anomaly detection is done by calculating the squared Euclidean distance between the input and output image. Squared error magnifies larger errors while disregards small ones.

3 Analysis Pipeline

The TLS-AAE was implemented as a part of an analysis pipeline for evaluating real-world performance and suitability. The implementation was made using open source software frameworks. The pipeline works in any network where the traffic can be mirrored for the analysis system for inspection. Figure 3 illustrates the architecture of the pipeline. The line consists of two main components, the preprocessing pipeline and the machine learning model.

3.1 Data processing

In the start of the pipeline, a slightly modified version of Suricata IDS software [23] was used for constructing network flows from the individual mirrored packets. The modification to Suricata software was made for collecting individual packet timings for each flow.

The main pipeline functionality was implemented using Apache Kafka [1] as a message queue. Suricata was configured to send the flows to one of the Kafka topics. From there, the flows are consumed using Apache Spark [2] platform running a custom preprocessing and feature extraction script. The extracted features are then sent back to Kafka, and delivered to the machine learning algorithm. In the final step of the pipeline the TLS-AAE returns the anomaly scores back to Kafka, after predicting the anomaly score.

The Apache Spark platform was used for extracting the needed features from flows. It was used for flow filtering, flow aggregation, and wavelet calculation by writing a custom script. The script filters all the flows that are not TLS traffic and aggregates the flows using a time window. The aggregation joins flows in a specified time window using source IP, destination IP, destination port and JA3 hash[6]. The JA3 hash is used in aggregation to separate different applications on a host.

The flows include the timing and size information for every packet. Packet timings are reduced from microseconds to seconds in accuracy to reduce unnecessary computational complexity. The packet timings and sizes are made into an impulse signal, where the received packets have negative values, sent packets positive values, the impulse values being the packet sizes. The signal is zero-padded from both ends so that the length of the signal is power of two and the

[6] https://github.com/salesforce/ja3

minimum length is reached. The minimum length of the signal depends on how many layers from the wavelet transformation are needed.

The wavelet transform is calculated for the signal using the Haar mother wavelet. Only the last N layers of the transform result are maintained so that all the results are of the same size regardless of the original signal length. The wavelet result is then transformed to an X x Y matrix. The transformation is made so that the results are easy to visualize. Absolute values are taken from the coefficients and the matrix values are normalized per matrix. The matrix is made in a way that each detail coefficient value populates equal amount of elements in the matrix. Figure 1 shows the matrix transformation visualized as an image. The color in the image represents the element value. The lower squares in the image represent the higher frequencies and the position from the left represents in which time the frequency has happened in the signal.

The results are sent back to Kafka where the machine learning algorithm can consume it and produce the anomaly score as the result.

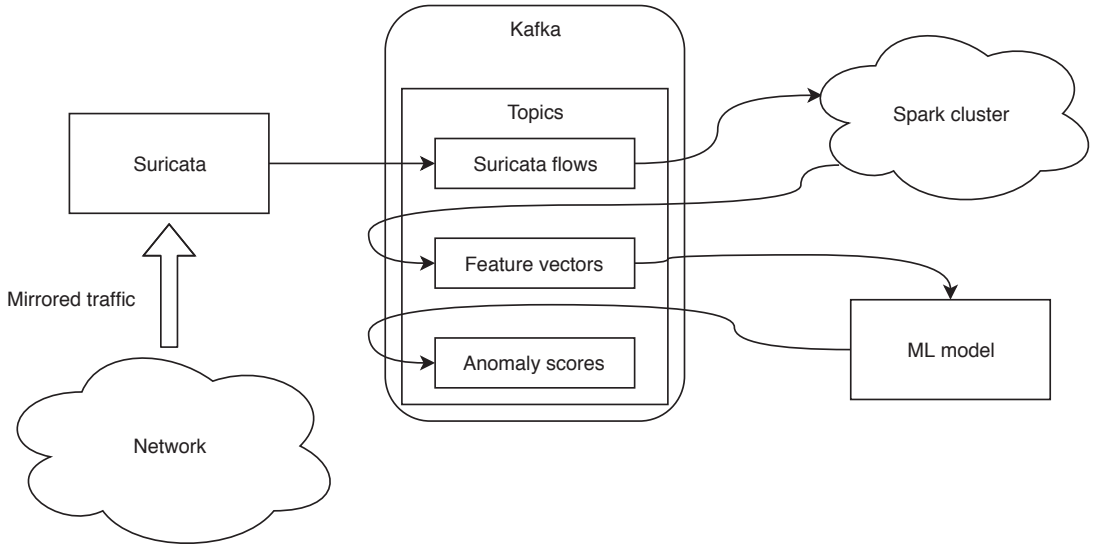


Fig. 3: Architecture of the implemented IDS solution

3.2 Dataset

A relevant and modern dataset is required for training, testing, and evaluating a IDS solution. Although some public datasets, such as the DARPA intrusion detection evaluation dataset [16] exists, they tend to be dated and not suitable for modern research. They lack both up-to-date cyber attacks, and modern traffic profile.

Since 2013 the Finnish national cyber security exercise has been conducted using the RGCE Cyber Range. In 2018 the exercise was organized by The Ministry of Defence, The Security Committee, and JAMK University of Applied

Sciences. The exercise is a large-scale live cyber security exercise, with more than 100 individuals from different national security authorities exercising co-operation during the cyber incidents. [18]

The data set used in this study was created from network traffic captured during the exercise. The whole traffic capture, at full packet level, consists of over 100,000,000 network flows from which a subset of 56,408,665 flows were captured from a place where anomaly traffic was present. This subset was used to create the training and testing data sets. The data set contains 729,998 flows that are TLS traffic, of which 665 flows are malicious.

The flows contain both human and auto-generated web browsing traffic, authentication portal logins, automatic updates of software, e-mail protocols that use TLS, and other common benign activity. Malicious flows were generated by Meterpreter, Empire, or CobaltStrike.

3.3 RGCE Cyber Range

The Realistic Global Cyber Environment (RGCE) is a holistic cyber range suitable for various tasks such as training, exercises, research and development. RGCE mimics the structure and traffic of real Internet. For example, ISP tiers are emulated using real hardware and structure. Network distances and latencies reflect those in real world, up to including packet losses. The network traffic of RGCE Cyber Range is generated according to a realistic end user traffic model, which augments the traffic generated by humans. RGCE includes industry specific organization environments, with complex deployments. For example, financial organization, electricity company and Internet service providers all have realistic AD infrastructure, SCADA systems, and other specialized production assets. [12, 11]

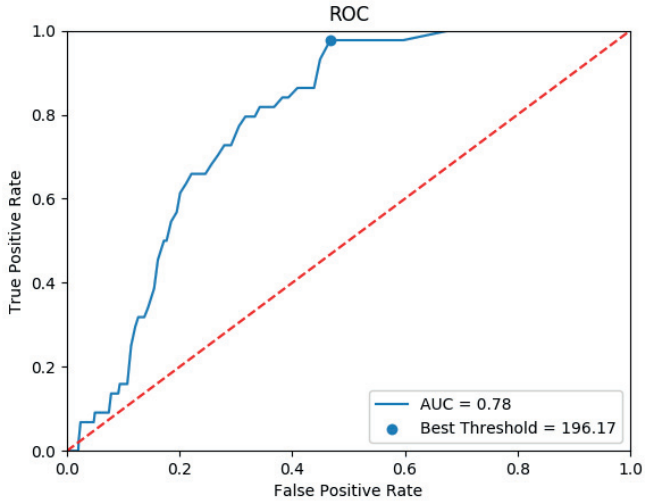
4 Results

The evaluation of the performance was made by using the receiver operating characteristic (ROC) curve by plotting the true positive rate (TPR) to y-axis and false positive rate (FPR) to x-axis [24]. Overall, the following characteristics were considered: TPR, FPR, and accuracy [24, 19].

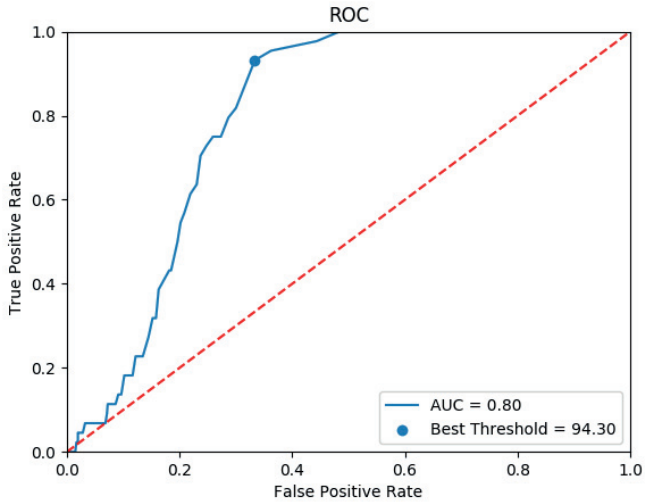
For evaluation of TLS-AAE, the prediction results were compared against a traditional autoencoder. The performance characteristics are listed in Table 1 and ROC curves are plotted in Figure 4.

Method	**TPR**	**FPR**	**Accuracy**
TLS-AAE	95%	36%	65%
Plain autoencoder	97%	47%	55%

Table 1: Performance characteristics



(a) Traditional autoencoder



(b) TLS-AAE

Fig. 4: Receiver operating characteristic curves for both plain autoencoder and the TLS-AAE. Best thresholds are based on the best ratio.

The results indicate that the TLS-AAE achieves similar TPR as the plain counterpart. However, the FPR is considerably lower, resulting in better accuracy. The input image of the autoencoder is 128x128 pixels and the grayscale values vary between 0 and 255 integers. Both autoencoders have two dense layers of 2048 units in both encoder and decoder. The bottleneck of the adversarial autoencoder consists of 10 dimensional latent and 20 dimensional cluster linearly activated variables. For consistency, the traditional autoencoders latent is 30 dimensional. The output of the decoder uses sigmoid activation to map the decoded values between 0 and 1.

5 Discussion

Based on the findings, the Haar wavelet transform seems to provide adequate representation on the nature of the TLS connections in the dataset, allowing categorization. Time window aggregation of distinct but related TLS flows captures malicious programs that infrequently poll the C2 server, opening a new TLS connection each time.

Considering that the connections were encrypted and only the size and timing information was available for analysis, the unsupervised TLS-AAE was able to construct a relatively representative latent space. Even though the dataset is relatively extensive in size, the variance of the flows is constricted. The relatively high false positive rate is partly explained by non-malicious outliers. The number of false positives can be drastically lowered by augmenting the results with other means of traffic analysis, such as IP / domain reputation.

The modified Suricata IDS, Spark, Kafka, and TensorFlow – in combination – proved to be a working base for an IDS solution. As Suricata can either process live mirrored traffic or replay an existing packet capture, developing models using the platform is relatively straightforward process.

6 Conclusion and Future Work

In this study Haar wavelet transforms and adversarial autoencoders were applied for constructing an anomaly detection based network intrusion detection system. For evaluation, a data pipeline based on open source software, including Suricata IDS, TensorFlow framework, Kafka message bus, and Spark framework, was constructed.

Network data from Finnish national cyber security exercise was used for the evaluation of the proposed model. The data was also used for finding and engineering suitable features for encrypted TLS connections. The test data included various attack vectors made by malware and exploitation frameworks.

Future work includes a more thorough statistical analysis on the TLS-AAE's latent space and its structure. Possible avenues of expansion are combining the current model with a more sophisticated predictor network. The wavelet transform and its applicability for TLS traffic analysis should be also further studied.

Acknowledgment

This research project is funded by MATINE - The Scientific Advisory Board for Defence.

References

1. Apache Kafka: Apache kafka A distributed streaming platform. `https://kafka.apache.org/`, accessed: 31 August 2018

2. Apache Spark: Apache Spark Lightning-fast unified analytics engine. https://spark.apache.org/, accessed: 31 August 2018
3. Chan, K., Fu, W.: Efficient time series matching by wavelets. In: Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337). pp. 126–133 (March 1999). https://doi.org/10.1109/ICDE.1999.754915
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. ACM Comput. Surv. **41**(3), 15:1–15:58 (Jul 2009). https://doi.org/10.1145/1541880.1541882
5. Daubechies, I.: The wavelet transform, time-frequency localization and signal analysis. IEEE Transactions on Information Theory **36**(5), 961–1005 (Sept 1990). https://doi.org/10.1109/18.57199
6. Dhingra, M., Jain, M., Jadon, R.S.: Role of artificial intelligence in enterprise information security: A review. In: 2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC). pp. 188–191 (Dec 2016). https://doi.org/10.1109/PDGC.2016.7913142
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 2672–2680. Curran Associates, Inc. (2014)
8. Haar, A.: Zur theorie der orthogonalen funktionensysteme. Mathematische Annalen **69**(3), 331–371 (Sep 1910). https://doi.org/10.1007/BF01456326
9. Hendler, D., Kels, S., Rubin, A.: Detecting Malicious PowerShell Commands Using Deep Neural Networks. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security. pp. 187–197. ASIACCS '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3196494.3196511
10. Husák, M., Čermák, M., Jirsík, T., Čeleda, P.: Https traffic analysis and client identification using passive ssl/tls fingerprinting. EURASIP Journal on Information Security **2016**(1), 6 (Feb 2016). https://doi.org/10.1186/s13635-016-0030-7
11. JAMK University of Applied Sciences, Institute of Information Technology, JYVSECTEC: Rgce cyber range. http://www.jyvsectec.fi/en/rgce/, accessed: 23 August 2018
12. Kokkonen, T., Puuska, S.: Blue Team Communication and Reporting for Enhancing Situational Awareness from White Team Perspective in Cyber Security Exercises. In: Galinina, O., Andreev, S., Balandin, S., Koucheryavy, Y. (eds.) Internet of Things, Smart Spaces, and Next Generation Networks and Systems. pp. 277–288. Springer International Publishing, Cham (2018)
13. Komar, M., Kochan, V., Dubchak, L., Sachenko, A., Golovko, V., Bezobrazov, S., Romanets, I.: High performance adaptive system for cyber attacks detection. In: 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). vol. 2, pp. 853–858 (Sept 2017). https://doi.org/10.1109/IDAACS.2017.8095208
14. Le, T., Kim, J., Kim, H.: An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization. In: 2017 International Conference on Platform Technology and Service (PlatCon). pp. 1–6 (Feb 2017). https://doi.org/10.1109/PlatCon.2017.7883684
15. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. pp. 2–11. DMKD '03, ACM, New York, NY, USA (2003). https://doi.org/10.1145/882082.882086

16. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyschogrod, D., Cunningham, R.K., Zissman, M.A.: Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation. In: Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00. vol. 2, pp. 12–26 vol.2 (Jan 2000). https://doi.org/10.1109/DISCEX.2000.821506
17. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.: Adversarial Autoencoders. In: International Conference on Learning Representations (2016), http://arxiv.org/abs/1511.05644
18. Ministry of Defence Finland: The national cyber security exercises is organised in Jyväskylä - Kansallinen kyberturvallisuusharjoitus kyha18 järjestetään Jyväskylässä, official bulletin 11th of may 2018. https://valtioneuvosto.fi/artikkeli/-/asset_publisher/kansallinen-kyberturvallisuusharjoitus-kyha18-jarjestetaan-jyvaskylassa (May 2018), accessed: 23 August 2018
19. Mokarian, A., Faraahi, A., Delavar, A.G.: False positives reduction techniques in intrusion detection systems-a review. IJCSNS International Journal of Computer Science and Network Security **13**(10), 128–134 (2013)
20. Pham, T.S., Hoang, T.H., Vu, V.C.: Machine learning techniques for web intrusion detection — A comparison. In: 2016 Eighth International Conference on Knowledge and Systems Engineering (KSE). pp. 291–297 (Oct 2016). https://doi.org/10.1109/KSE.2016.7758069
21. Rescorla, E., Dierks, T.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (August 2008). https://doi.org/10.17487/RFC5246
22. Sommer, R., Paxson, V.: Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In: 2010 IEEE Symposium on Security and Privacy. pp. 305–316 (May 2010). https://doi.org/10.1109/SP.2010.25
23. Suricata: Suricata Open Source IDS / IPS / NSM engine. https://suricata-ids.org/, accessed: 31 August 2018
24. Suyal, P., Pant, J., Dwivedi, A., Lohani, M.C.: Performance evaluation of rough set based classification models to intrusion detection system. In: 2016 2nd International Conference on Advances in Computing, Communication, Automation (ICACCA) (Fall). pp. 1–6 (Sept 2016). https://doi.org/10.1109/ICACCAF.2016.7748991
25. Vartouni, A.M., Kashi, S.S., Teshnehlab, M.: An anomaly detection method to detect web attacks using stacked auto-encoder. In: 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS). pp. 131–134 (Feb 2018). https://doi.org/10.1109/CFIS.2018.8336654

P6

NETWORK ANOMALY DETECTION BASED ON WAVENET

by

T. Kokkonen, S. Puuska, J. Alatalo, E. Heilimo & A. Mäkelä 2019

In: Galinina O., Andreev S., Balandin S., Koucheryavy Y. (eds) Internet of Things, Smart Spaces, and Next Generation Networks and Systems. NEW2AN 2019, ruSMART 2019. Lecture Notes in Computer Science, vol 11660 pp. 424–433. Springer, Cham

https://doi.org/10.1007/978-3-030-30859-9_36

Network Anomaly Detection based on WaveNet

Tero Kokkonen, Samir Puuska, Janne Alatalo, Eppu Heilimo, and Antti Mäkelä

Institute of Information Technology, JAMK University of Applied Sciences,
Jyväskylä, Finland
{tero.kokkonen, samir.puuska, janne.alatalo, eppu.heilimo,
antti.mäkelä}@jamk.fi

Abstract. Increasing amount of attacks and intrusions against networked systems and data networks requires sensor capability. Data in modern networks, including the Internet, is often encrypted, making classical traffic analysis complicated. In this study, we detect anomalies from encrypted network traffic by developing an anomaly based network intrusion detection system applying neural networks based on the WaveNet architecture. Implementation was tested using dataset collected from a large annual national cyber security exercise. Dataset included both legitimate and malicious traffic containing modern, complex attacks and intrusions. The performance results indicated that our model is suitable for detecting encrypted malicious traffic from the datasets.

Keywords: Intrusion Detection · Anomaly Detection · WaveNet · Convolutional Neural Networks

1 Introduction

Intrusion detection systems (IDS) are divided into two categories: anomaly-based detection (anomaly detection) and signature-based detection (misuse detection). Anomaly-based-detection can be applied without pre-recorded signatures for unknown attack patterns and even for encrypted network traffic, however the weakness for anomaly detection is the high amount of false positive detections [3, 13].

Machine learning techniques have recently been applied successfully to network anomaly detection and classification [6]. Bitton and Shabtai in [1] have studied machine learning based IDS for Remote Desktop Protocols (RPD). Different machine learning techniques have been applied, e.g. Wiewel and Yang used Variational Autoencoder in their study [28], Chen et al. used Convolutional Autoencoder [2] while Long Short-Term Memory (LSTM) and Gated Recurrent Unit methods are used in the paper [6]. Paper [23] presents technique for increasing detection accuracy with feedback.

In our earlier study [19], we used Haar wavelet transforms and Adversarial Autoencoders (AA) [10] for implementing unsupervised network anomaly detection based IDS. Our earlier model, described in [19], had reasonable good operational characteristics; in this study we strived to improve it using alternative modeling approach. As argument of efficiency, numerical results are compared with the earlier results using the same dataset from Finland's National

Cyber Security Exercise [12]. Performance characteristics are also accomplished using publicly available reference intrusion detection evaluation dataset (CICIDS2017) [27].

Our study presents state-of-the-art network anomaly detection based intrusion detection system that exploits deep learning method WaveNet [15]. First, in section 2, this paper describes implemented anomaly detection method including feature extraction and analysis method. Then, in section 3, we introduce experimental results for the performance characteristics of our model and finally there are conclusions with found future research topics.

2 Anomaly Detection Method

2.1 Dataset

According to Nevavuori and Kokkonen [14], a network anomaly detection data set must (i) include network traffic data and (ii) host activity data, (iii) multiple scenarios, (iv) be representative of real-world circumstances, and (v) the format of the data must be usable.

Since many publicly available datasets already exist [20], we decided to utilize them in this research. Although notable public datasets, such as the KDD99 [25] and DARPA datasets [7–9] exist and are used in many existing network intrusion detection research, they are very old, and many researches have directed a lot of criticism against them [11, 24]. The main problem is that datasets do not include modern threat and attack patterns with required statistical characteristics nor sophisticated and modern architectures [14, 26, 4, 22]. In many datasets the raw data is already processed into network flows losing the information of individual packet timings. Fortunately, in addition to the processed flow data, some datasets include the raw packet captures.

The Intrusion Detection Evaluation Dataset (CICIDS2017) by the Canadian Institute for Cybersecurity [27] is one of the more modern publicly available datasets. Although the dataset was created with a traffic generator, it was modeled after modern real-world network traffic. It includes benign HTTPS network traffic and therefore is suitable for research concerning encrypted communication. Unfortunately, the dataset does not include many TLS based attacks, which form a sizable amount of modern malware control channels.

We decided to use the benign traffic from the CICIDS2017 dataset as clean traffic during the model development and testing, but because the anomalous traffic in the dataset was not large enough, more anomalous traffic was required. We generated additional anomalous traffic in our own environment using Empire PowerShell post-exploitation agent [1] and Cobalt Strike [2]; both are adversary simulation frameworks that use real-world malware characteristics. A small amount of benign traffic was also generated in the environment. The benign traffic was

[1] https://www.powershellempire.com/
[2] https://www.cobaltstrike.com/

generated by controlling Windows virtual machine using a scripted bot that operated normal GUI software with virtual mouse and keyboard aided by computer vision. This data was used in the evaluation to make sure that the environments are compatible enough so that our generated benign traffic is not classified as anomalous with the model that is trained with the CICIDS2017 benign data.

In addition to the CICIDS2017 dataset and the self-generated dataset, the final model was also tested with the Finland's National Cyber Security Exercise dataset (FNCSE2018), also used in our previous publication [19]. This dataset was used to get comparable results to our previous research. RGCE Cyber Range (Realistic Global Cyber Environment) is used for research and development or training and exercises. In the RGCE Cyber Range main structures and services of the real Internet are modeled with the realistic user traffic patterns of users. RGCE offers tailored organization environments with real assets [5]. Finland's National Cyber Security Exercise is conducted annually in the RGCE Cyber Range. Network data from the real Cyber Security Exercise conducted in the RGCE Cyber Range includes realistic complex environment and legitimate network traffic mixed with modern attack patterns for testing the capabilities of Intrusion Detection System capability. [12] In this study we were authorized to use the traffic captures from Finland's National Cyber Security Exercise of 2018.

2.2 Feature Extraction

Our research focused on finding the anomalies based on packet timing patterns. This choice was made to accommodate encrypted command and control channels modern malware use. Traditional deep inspection techniques and statistical analyses that utilize payloads are incompatible with modern security landscape, made e.g. decrypting proxies obsolete due to various certificate pinning features. In this project we used a modified version of Suricata IDS software [18] to process the raw packet capture files into parsed network data. The modification in the software allowed the packet timings information to be extracted from packet capture files along with the parsed data.

The CICIDS2017 dataset includes the raw packet captures in addition to labeled processed flow data. Since the processed flow data does not include packet timings, the raw data had to be reprocessed to flow data with the modified Suricata software. The processed flows were then labeled by joining the flows to the CICIDS2017 flow labels by matching flow timestamps, IP addresses and network ports. The result was labeled flows from the CICIDS2017 dataset including packet timings. Because our system used different software for packet capture to network flow conversion from the one used in CICDS2017, the resulting flows did not match exactly, resulting in lost flows. Only the flows that matched correctly between Suricata processed flows and CICDS2017 labeled flows were retained in the dataset. Based on the flow label, the dataset was then split to anomaly and benign flows. All the flows that did not have *benign* label were treated as anomalies. The final processed CICDS2017 dataset included 1,425,742 flows, of which 1,107,695 were labeled as benign flows, and 318,047 flows were labeled as

non-benign flows. From the 1,107,695 benign flows, 307,771 were TLS flows. Originally the Suricata processed CICIDS2017 packet capture files included 1,956,363 flows, so 530,621 flows did not find matching flow in the CICIDS2017 flow label files. This can be almost certainly accounted on the poor quality of the flow label files in CICIDS2017 dataset. The files include a duplicate entry for most of the flows and the flow timestamps are recorded in a minute accuracy with an ambiguous 12-hour clock format.

The FNCSE2018 dataset and our self generated datasets were processed in the same way. The labels were assigned by hand based on known origin and destination addresses of the attacks. The FNCSE2018 dataset included 715,158 benign TLS flows, and 653 non-benign TLS flows. The self generated dataset included 15,124 benign flows and 7,991 non benign flows.

The resulting flows were then further processed by calculating timing differences between packets. The final features for one packet in a flow were: *packet direction*, *time difference to next received packet*, *time difference to next transmitted packet* and *packet size*. The timing differences varied from microseconds to minutes with most of the differences being very small. Because our model required quantization of the input data, the timing differences were scaled with the common logarithm to better utilize the reduced quantization precision. The packet sizes were scaled in similar way for the same reason. This choice is warranted, because in network traffic large delays are often the result of an unrelated problem, and not an inherent feature of the protocol in question. Although many protocols, including malware command channels, may use delays and timers, there usually is no reason to keep using the same flow. Packet sizes follow the same scaling principle, the maximum size being the MTU of the path. Small packet sizes and the variation therein are are likely to be indicative of the intrinsic properties of the protocol, unlike the variation near the MTU. This is especially apparent in many malware communication protocols, which often use fixed size binary messages. The aforementioned adversary simulation frameworks also exhibit this phenomenon.

2.3 Multi-feature WaveNet

The network traffic was analyzed with a deep neural network model based on the WaveNet [15] architecture, illustrated in the Figure 2. WaveNet was chosen as a basis for our model for its capability to directly interface with variable length sequential data. This enables us to feed complete and unreduced sequences to the model. We utilized this trait to predict network traffic connections of varying length packet by packet.

The primary task of the model is to predict the next sample by using prior samples. The core network structure consists of a variation of the WaveNet architecture configured for multiple features. The modified WaveNet is extended to utilize two-dimensional dilated causal convolutions; input data is arranged into a two-dimensional lattice, discrete time steps forming the first dimension and individual sample features along the other dimension. Dilated convolutions expand the receptive field of the network exponentially [29], giving the model a

potential to observe long term temporal dependencies. Dilation of convolutions is only performed along the time axis of the data, as the receptive fields are exceedingly large and thus not optimal for the relatively small fixed length feature axis. The causality aspect of the convolutions is used to assert an ordered time-dependency on the input data: predicted samples may only depend on preceding input samples. We implemented the causality by padding the beginning of the sequence by the filter size in the first layer and by (filter size$-1)\times$dilation rate in the subsequent layers, effectively shifting the convolution operations. The causal layer stack is visualized in Figure 1.

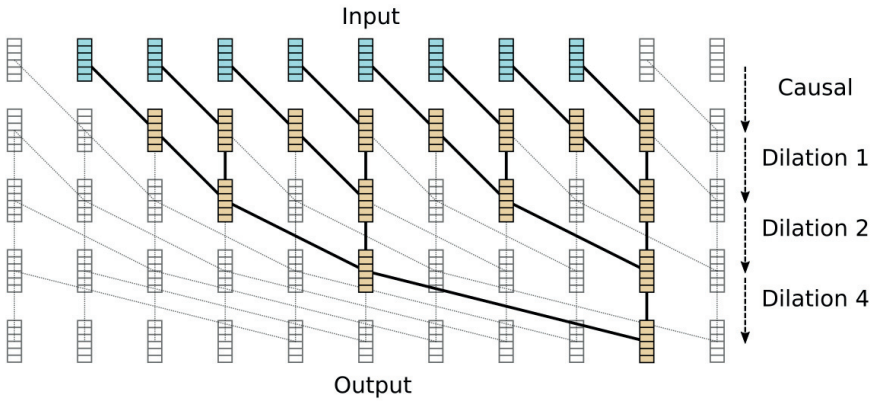


Fig. 1: Visualization of the models two-dimensional dilated causal layers and the first causal layer.

The input variables are quantized to n bins, continuous and discrete variables alike, matching the practice used in WaveNet [15] as well as PixelRNN [16]. As the length of the input data varies with each example, a special end of sequence value is used to represent sequence termination. The network utilizes a discretized mixture of logistic distributions, as described in PixelCNN++ [21] and Parallel WaveNet [17]. We found this to perform slightly better when compared to a more classical soft-max layer.

The individual residual layers follow closely the structure present in WaveNet. Unlike the WaveNet architecture, we included a dropout layer before each dilated convolution layer as shown in Figure 2. Applying dropout inside each residual layer has been previously explored in PixelCNN++ [21] and Wide Residual Networks [30].

To distinguish anomalous data from benign data, an anomaly score is quantified from the network outputs with a single forward pass, effectively avoiding the downside of slow sampling of the WaveNet model. In our approach, we computed the training loss contributions for each sample in the input sequence. The overall anomaly score of the whole sequence was the mean of these loss values, with samples past the end of sequence marker masked out to account for different length of sequences.

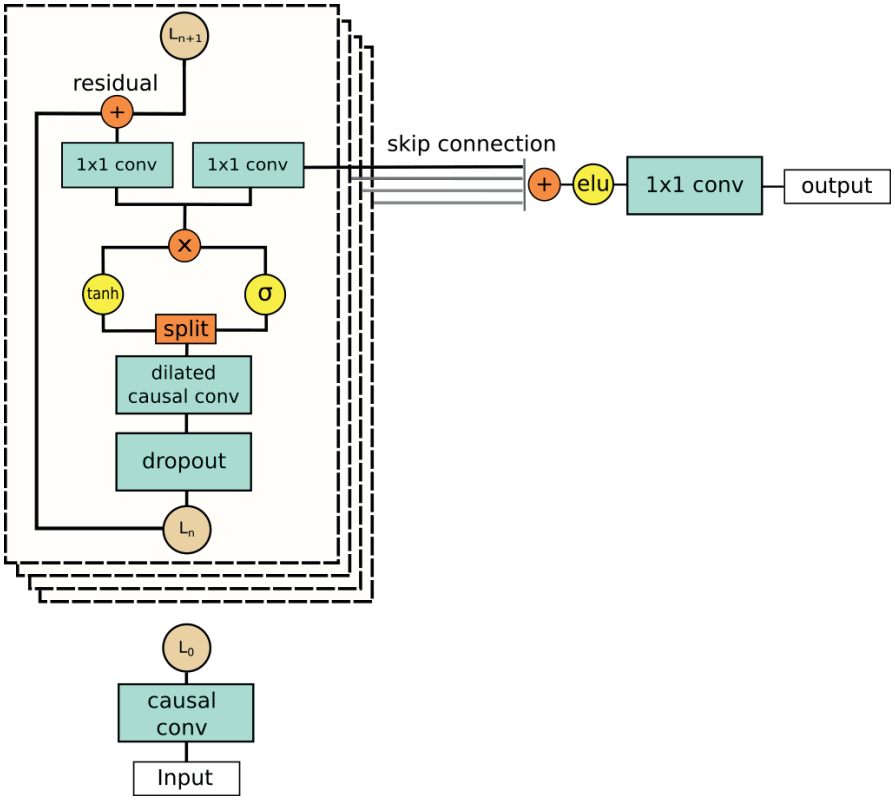


Fig. 2: The architecture is similar to the original WaveNet [15], with the exception of a dropout layer between all dilation layers and exclusive weights between residual and skip connections.

3 Experimental Results

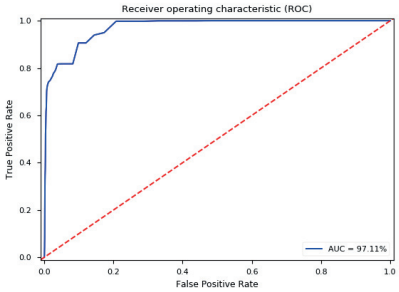
For the numerical results, we created receiver operating characteristic (ROC) curves by plotting the true positive rate (TPR) to y-axis and false positive rate (FPR) to x-axis. As a comparable score we also calculated the area under curve (AUC) from the ROC.

Training Dataset	Evaluation Dataset	AUC
CICIDS2017	CICIDS2017	97.11%
CICIDS2017	Our TLS anomalies	99.48%
CICIDS2017	CICIDS2017 + Our TLS anomalies	96.81%
FNCSE2018	FNCSE2018	91.61%

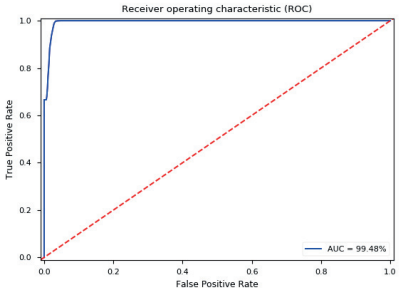
Table 1: Area under curve scores for four different evaluation dataset combinations.

In order to model an anomaly detector we split the clean data from CICIDS2017 and FNCSE2018 datasets into training and evaluation parts using

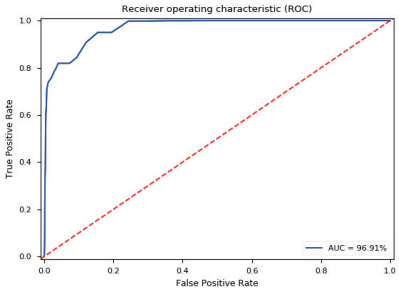
80/20 ratio. We took 256 first packets from each flow and trained a model with 9 dilation layers (receptive field of 256), vertical filter size of 3 and horizontal 2, 128 filters each layer for $\sim$ 15 epochs while evaluating the model using the evaluation part of the dataset to keep the model from over-fitting. During and after the training we ran an evaluation where we included the anomaly data to validate the anomaly detection capability of the model. Since the CICIDS2017 dataset lacks TLS anomalies we ran the evaluation three times to validate the model against the included CICIDS2017 anomalies, our TLS anomalies and a mixture of both. The resulting AUC scores are listed in Table 1. The FNCSE2018 training and evaluation datasets include only TLS encrypted connections.



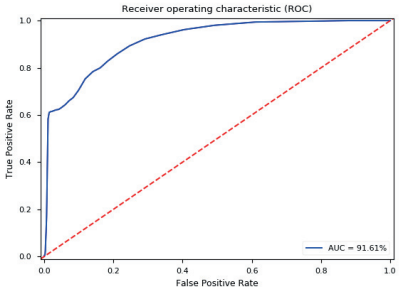
(a) CICIDS2017 Anomalies



(b) Our TLS Anomalies



(c) CICIDS2017 and TLS Anomalies



(d) FNCSE2018

Fig. 3: Receiver operating characteristic curves on the four datasets we used to evaluate the model.

From the results in Figure 3 we concluded that the model is capable of detecting anomalies in both datasets, while also retaining the capability of detecting anomalous connection with TLS encryption. The model also performs significantly better than our earlier model [19], which had 80% AUC whereas the new model got 91.61% AUC on the same dataset.

4 Conclusion

In this study we applied the WaveNet and PixelCNN models for constructing an IDS based on anomaly detection. For the feature extraction and data processing, an open source software -based data pipeline was constructed. We utilized network data from Finland's National Cyber Security Exercise as well as public reference dataset CICIDS2017. The combined dataset was relatively extensive, although further efforts should be made to include a more diverse selection of applications and web browsing activities.

Results suggest that the machine learning model is suitable for detecting malicious command and control channels from TLS encrypted connections. The model is able to circumvent issues arising from samples of various lengths, and quantize timing and packet size differences into ranges suitable for neural networks.

Future work includes a conditioned WaveNet, variational or adversarial encoder to self-condition the WaveNet, and further testing on possible anomaly scores. Furthermore, visualization methods of found network anomalies should be studied for achieving better situational awareness in operative environments.

Acknowledgment

This research project is funded by MATINE - The Scientific Advisory Board for Defence.

References

1. Bitton, R., Shabtai, A.: A Machine Learning-Based Intrusion Detection System for Securing Remote Desktop Connections to Electronic Flight Bag Servers. IEEE Transactions on Dependable and Secure Computing pp. 1–1 (2019). https://doi.org/10.1109/TDSC.2019.2914035
2. Chen, Z., Yeo, C.K., Lee, B.S., Lau, C.T.: Autoencoder-based network anomaly detection. In: 2018 Wireless Telecommunications Symposium (WTS). pp. 1–5 (April 2018). https://doi.org/10.1109/WTS.2018.8363930
3. Chiba, Z., Abghour, N., Moussaid, K., Omri, A.E., Rida, M.: A Clever Approach to Develop an Efficient Deep Neural Network Based IDS for Cloud Environments Using a Self-Adaptive Genetic Algorithm. In: 2019 International Conference on Advanced Communication Technologies and Networking (CommNet). pp. 1–9 (April 2019). https://doi.org/10.1109/COMMNET.2019.8742390
4. Creech, G., Hu, J.: Generation of a new IDS test dataset: Time to retire the KDD collection. In: IEEE Wireless Communications and Networking Conference, WCNC. pp. 4487–4492. IEEE (apr 2013). https://doi.org/10.1109/WCNC.2013.6555301
5. JAMK University of Applied Sciences, Institute of Information Technology, JYVSECTEC: Rgce cyber range. http://www.jyvsectec.fi/en/rgce/, accessed: 26 April 2019

6. Li, Z., Rios, A.L.G., Xu, G., Trajković, L.: Machine Learning Techniques for Classifying Network Anomalies and Intrusions. In: 2019 IEEE International Symposium on Circuits and Systems (ISCAS). pp. 1–5 (May 2019). https://doi.org/10.1109/ISCAS.2019.8702583
7. Lincoln Laboratory, Massachusetts Institute of Technology: 1998 DARPA Intrusion Detection Evaluation Dataset. https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, accessed: 29 April 2019
8. Lincoln Laboratory, Massachusetts Institute of Technology: 1999 DARPA Intrusion Detection Evaluation Dataset. https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset, accessed: 29 April 2019
9. Lincoln Laboratory, Massachusetts Institute of Technology: 2000 DARPA Intrusion Detection Scenario Specific Datasets. https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets, accessed: 29 April 2019
10. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.: Adversarial Autoencoders. In: International Conference on Learning Representations (2016), http://arxiv.org/abs/1511.05644
11. McHugh, J.: Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations As Performed by Lincoln Laboratory. ACM Trans. Inf. Syst. Secur. **3**(4), 262–294 (Nov 2000). https://doi.org/10.1145/382912.382923, http://doi.acm.org/10.1145/382912.382923
12. Ministry of Defence Finland: The national cyber security exercises is organised in Jyväskylä - Kansallinen kyberturvallisuusharjoitus kyha18 järjestetään Jyväskylässä, official bulletin 11th of may 2018. https://valtioneuvosto.fi/artikkeli/-/asset_publisher/kansallinen-kyberturvallisuusharjoitus-kyha18-jarjestetaan-jyvaskylassa (May 2018), accessed: 26 April 2019
13. Narsingyani, D., Kale, O.: Optimizing false positive in anomaly based intrusion detection using Genetic algorithm. In: 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE). pp. 72–77 (Oct 2015). https://doi.org/10.1109/MITE.2015.7375291
14. Nevavuori, P., Kokkonen, T.: Requirements for Training and Evaluation Dataset of Network and Host Intrusion Detection System. In: Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S. (eds.) New Knowledge in Information Systems and Technologies. pp. 534–546. Springer International Publishing, Cham (2019)
15. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio (2016), https://arxiv.org/pdf/1609.03499.pdf
16. van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel Recurrent Neural Networks. In: Balcan, M.F., Weinberger, K.Q. (eds.) Proceedings of The 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 1747–1756. PMLR, New York, New York, USA (20–22 Jun 2016), http://proceedings.mlr.press/v48/oord16.html
17. van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L.C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., Hassabis, D.: Parallel WaveNet: Fast High-Fidelity Speech Synthesis. CoRR **abs/1711.10433** (2017), http://arxiv.org/abs/1711.10433
18. Open Information Security Foundation (OISF): Suricata Open Source IDS / IPS / NSM engine. https://suricata-ids.org/, accessed: 7 May 2019

19. Puuska, S., Kokkonen, T., Alatalo, J., Heilimo, E.: Anomaly-Based Network Intrusion Detection Using Wavelets and Adversarial Autoencoders. In: Lanet, J.L., Toma, C. (eds.) Innovative Security Solutions for Information Technology and Communications. pp. 234–246. Springer International Publishing, Cham (2019)
20. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A.: A Survey of Network-based Intrusion Detection Data Sets. Computers & Security **86**, 147 – 167 (2019). https://doi.org/10.1016/j.cose.2019.06.005
21. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017 (2017), https://openreview.net/references/pdf?id=rJuJ1cP_l
22. Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A.: Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers and Security **31**(3), 357–374 (may 2012). https://doi.org/10.1016/j.cose.2011.12.012
23. Siddiqui, M.A., Stokes, J.W., Seifert, C., Argyle, E., McCann, R., Neil, J., Carroll, J.: Detecting Cyber Attacks Using Anomaly Detection with Explanations and Expert Feedback. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2872–2876 (May 2019). https://doi.org/10.1109/ICASSP.2019.8683212
24. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications. pp. 53–58. CISDA'09, IEEE Press, Piscataway, NJ, USA (2009), http://dl.acm.org/citation.cfm?id=1736481.1736489
25. The University of California Irvine (UCI): KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed: 29 April 2019
26. Umer, M.F., Sher, M., Bi, Y.: Flow-based intrusion detection: Techniques and challenges. Computers and Security **70**, 238–254 (2017). https://doi.org/10.1016/j.cose.2017.05.009
27. University of New Brunswick, Canadian Institute for Cybersecurity: Intrusion Detection Evaluation Dataset (CICIDS2017). https://www.unb.ca/cic/datasets/ids-2017.html, accessed: 30 April 2019
28. Wiewel, F., Yang, B.: Continual Learning for Anomaly Detection with Variational Autoencoder. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3837–3841 (May 2019). https://doi.org/10.1109/ICASSP.2019.8682702
29. Yu, F., Koltun, V.: Multi-Scale Context Aggregation by Dilated Convolutions. CoRR **abs/1511.07122** (2016), https://arxiv.org/pdf/1511.07122.pdf
30. Zagoruyko, S., Komodakis, N.: Wide Residual Networks. In: Richard C. Wilson, E.R.H., Smith, W.A.P. (eds.) Proceedings of the British Machine Vision Conference (BMVC). pp. 87.1–87.12. BMVA Press (September 2016). https://doi.org/10.5244/C.30.87

P7

STATISTICAL EVALUATION OF ARTIFICIAL INTELLIGENCE -BASED INTRUSION DETECTION SYSTEM

by

S. Puuska, T. Kokkonen, P. Mutka, J. Alatalo, E. Heilimo & A. Mäkelä 2020

In: Rocha Á., Adeli H., Reis L., Costanzo S., Orovic I., Moreira F. (eds)
Trends and Innovations in Information Systems and Technologies
WorldCIST 2020. Advances in Intelligent Systems and Computing,
vol 1160 pp. 464–470. Springer, Cham.

https://doi.org/10.1007/978-3-030-45691-7_43

Statistical Evaluation of Artificial Intelligence -based Intrusion Detection System

Samir Puuska, Tero Kokkonen, Petri Mutka, Janne Alatalo, Eppu Heilimo, and Antti Mäkelä

Institute of Information Technology, JAMK University of Applied Sciences, Jyväskylä, Finland
{samir.puuska, tero.kokkonen, petri.mutka, janne.alatalo, eppu.heilimo, antti.makela}@jamk.fi

Abstract. Training neural networks with captured real-world network data may fail to ascertain whether or not the network architecture is capable of learning the types of correlations expected to be present in real data.
In this paper we outline a statistical model aimed at assessing the learning capability of neural network-based intrusion detection system. We explore the possibility of using data from statistical simulations to ascertain that the network is capable of learning so called precursor patterns. These patterns seek to assess if the network can learn likely statistical properties, and detect when a given input does not have those properties and is anomalous.
We train a neural network using synthetic data and create several test datasets where the key statistical properties are altered. Based on our findings, the network is capable of detecting the anomalous data with high probability.

Keywords: Statistical Analysis, Intrusion Detection, Anomaly Detection, Network Traffic Modeling, Autoregressive Neural Networks

1 Introduction

Neural networks are being increasingly used as a part of Intrusion Detection Systems, in various configurations. These networks are often trained in ways that include both legitimate and malicious recorded network traffic. Traditionally, a training set is used to train the network, while another set of samples is used to assess the suitability of the proposed architecture. However, further assessment of the network architecture depends on knowing what statistical properties the network can learn, and how it will react if these properties change.

In this paper, we present a way to estimate if a network has the capability of learning certain desired features. Our analysis approach is to ascertain that the network can learn precursor patterns, i.e. patterns that are necessary but not sufficient conditions for learning more complex patterns of the same type. The goal is to supplement traditional sample-based learning with synthetic data

variants that have predictable and desirable statistical properties. This synthetic data can then be used both to increase the dataset and to address known biases that often arise when collecting real-world data traffic.

Certain real-life phenomena, such as network traffic, can be considered to have known intrinsic properties due to their artificial nature. In communication protocols, for example, certain hard limits must be observed for achieving any successful communication. Although protocols are sometimes abused for malicious purposes, there are still limits as to how extensive the effect can realistically be. On other occasions, there are limits on how much any given feature can be expected to correlate with anomalies. However, a combination of these weakly-correlated features may, if they form a specific pattern, signal for an anomaly. In artificial systems, it is sometimes possible to distinguish correlation from causation, and therefore make more intelligent predictions by considering only the direction that is actually feasible.

Based on their basis of analysis, there are two classes of Intrusion Detection Systems (IDS): anomaly-based detection (anomaly detection) and signature-based detection (misuse detection). Anomaly-based detection functions without earlier gathered signatures and are effective even for zero-day attacks and encrypted network traffic. There are various machine learning techniques implemented for classifying anomalies from network traffic but still, some flaws exist; a high amount of false alarms and low throughput [2, 6, 5].

In our earlier studies, we implemented two anomaly-detection based IDSs that utilized deep learning. Our first model was based on wavelet transforms and Adversarial Autoencoders [8]. That model was improved with a WaveNet [7] based solution [4]. In this paper, we perform a statistical experiment for determining the performance of a WaveNet based IDS system.

2 Method

We begin by outlining a statistical model which complies with our research goals. As stated, the idea is to construct a statistical distribution which contains so-called precursor or proto-elements of the actual phenomenon. The aim here is to ascertain that the network is capable of learning simpler versions of the relationships expected to be present in the real data.

Network protocols have a certain degree of predictability. As previously stated, we can state certain hard limits for the features we have selected. Our model is designed to work with the Transport Layer Security (TLS) protocols, as encrypted HTTP traffic is a common communication channel for malware.

We can identify various types of noise that usually occurs in the networks. The model should be resistant to this type of noise, as we know it arises due to the nature of data networks and is likely not associated with the type of anomalies we are interested in.

Based on this reasoning we have constructed a model that incorporates three distributions modeling i) packet size, ii) packet direction, and iii) packet timings.

One packet is modeled using these three features. The packet structure is illustrated in Figure 1. A connection consists of 250 packets (vectors), where timings are expressed using time differences to the next packet.

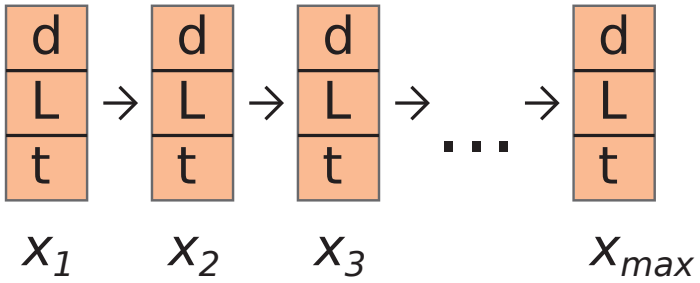


Fig. 1: Visual description of a single connection. Each packet consists of a vector that contains three elements: packet direction, packet size, and time difference to the next packet.

2.1 Packet size and direction

Based on the findings by Castro et al. [1], we model the packet size using the Beta distribution ($\alpha = 0.0888$, $\beta = 0.0967$). We enforce two strict cut off points: the minimum (15) and maximum (1500). This reflects the packet size constraints that networking protocols impose on packet size.

Packet direction is determined using the packet size. This models the real-world phenomenon where the requests are usually smaller than the responses. In the model packet direction, there is a binary value determined by packet size L; packets smaller than 30 are outgoing and larger than 1200 are incoming. If the size is $30 < L < 1200$, the direction is decided randomly.

2.2 Packet timing

Various separate processes affect packet timing: the nature of the protocol or data transfer type determines how fast packets are expected to be sent or received. For example, fetching an HTML page via HTTP creates a burst of packets going back and forth; however, malware that polls a Command and Control server at late intervals (for example hourly) may send just one packet and get one in response. However, a considerable amount of variance is expected when systems are under a high load or there is a network issue. Therefore, not all anomalies in the timing patterns are malicious in nature.

Since we do not need to model the traffic explicitly, we use a packet train model [3] inspired composite Gaussian distribution model for creating packet timings. Originally, the packet train model was designed for categorizing real-life network traffic, not for generating synthetic network data.

For the relevant parts, the packet train model is characterized by the following parameters; *mean inter-train arrival time*, *mean inter-car arrival time*, *mean train-size*. We capture the similar behavior by combining two normal probability density functions in range $x \in [a, b]$ as:

$$f(x) \equiv f(x; \mu_1, \mu_2, \sigma_1, \sigma_2, w_1, w_2, a, b) = \begin{cases} 0 & \text{if } x < a \\ \frac{R}{\sqrt{2\pi}} \left[\frac{w_1}{\sigma_1^2} \exp\left(-\frac{(x-\mu_1)^2}{2\sigma_1^2} \right) + \frac{w_2}{\sigma_2^2} \exp\left(-\frac{(x-\mu_2)^2}{2\sigma_2^2} \right) \right] & \text{if } a \leq x \leq b \\ 0 & \text{if } x > b \end{cases}, \quad (1)$$

where R is normalization constant that is calculated from normalization condition for total probability. In (1), μ_1 and μ_2 are mean values for sub-distributions ($\mu_1 < \mu_2$) , and σ_1 and σ_2 are relevant variances. Sub-distributions have relative weights w_1 and w_2.

We chose to use a semi-analytical probabilistic model since it is easier to parameterize and understand than more generic Markov models. Our model captures the most relevant properties of the train packet model; roughly *mean inter-train arrival time* $\propto \mu_2$, *mean inter-car arrival time* $\propto \mu_1$, and *mean train size* $\propto w_1/w_2$. Corresponding cumulative distribution function can be expressed with complementary error functions and solved numerically for generating random number samples with desired statistical properties.

2.3 Scoring

Since our neural network is trained by minimizing the mean of minibatches discretized logistic mixture negative log-likelihoods [9], we can detect the anomalous connections by observing the mean negative log-likelihood of the feature vectors in a single sample. Moreover, we introduce the different types of anomalies in varying quantities to the dataset to evaluate the neural network's sensitivity and behavior.

2.4 Tests

We trained the neural network using data formed by previously described clean distributions. The size of the training set was 160000 samples. We reserved an additional 40000 unseen samples for the evaluation.

The test procedure consists of generating samples where the parameters are drawn from a different distribution than the training data. These "anomalous" samples are mixed with the evaluation data to form ten sets where the percentage is increased from 10% to 100%. Each of these datasets is evaluated using the neural network and the changes in the mean anomaly score are observed in Table 1, which describes the three types of alterations made to the samples. The alterations were chosen because they represent different correlations; namely, the directionality is determined between two features inside one packet, whereas

the change in timing distribution is spread out between packets and does not correlate with other features inside a particular vector. This approach is expected to test the network's capability to detect both kinds of correlations.

Test	Description
Direction	This test swaps the directionality decision criteria. Small packages are now incoming and large outgoing. The area where the directionality is randomly determined stays the same.
Time	This test replaces the bimodal distribution on packet timing with unimodal Gaussian distribution $\mu = 50$, $\sigma = 80$. The cut-off points remain the same.
Combined	The test combines both alterations to the dataset.

Table 1: Descriptions of the alterations.

3 Results

The results indicate that the network learned to detect anomalous data in all three datasets. The results are illustrated in Figure 2. As the figure indicates, the anomaly score keeps increasing with the percentage of "anomalous" data.

Packet direction seems to have an almost linear increase in the anomaly score, whereas changes in time distribution result in a sudden jump, after which the score keeps increasing relatively modestly. The combined data exhibits both the starting jump and the linear increase. This is a desired outcome, as it indicates that the anomaly score reflects the change in data in a stable fashion.

In summary, the network was able to learn the properties outlined in the previous sections. The results indicate that the network can detect correlation inside the vector, as well as between vectors. This outcome supports the notion that a neural network structured in this fashion learns useful relationships between the features.

4 Discussion

When constructing a machine learning solution for anomaly detection, the available data may not be suitably representative. This situation may arise, for example, when collecting or sampling the dataset in a statistically representative way is impossible for practical reasons. It is not feasible to expect a statistically representative sample of all possible network flows, even when dealing with one application. Moreover, the data in networks may exhibit correlations known

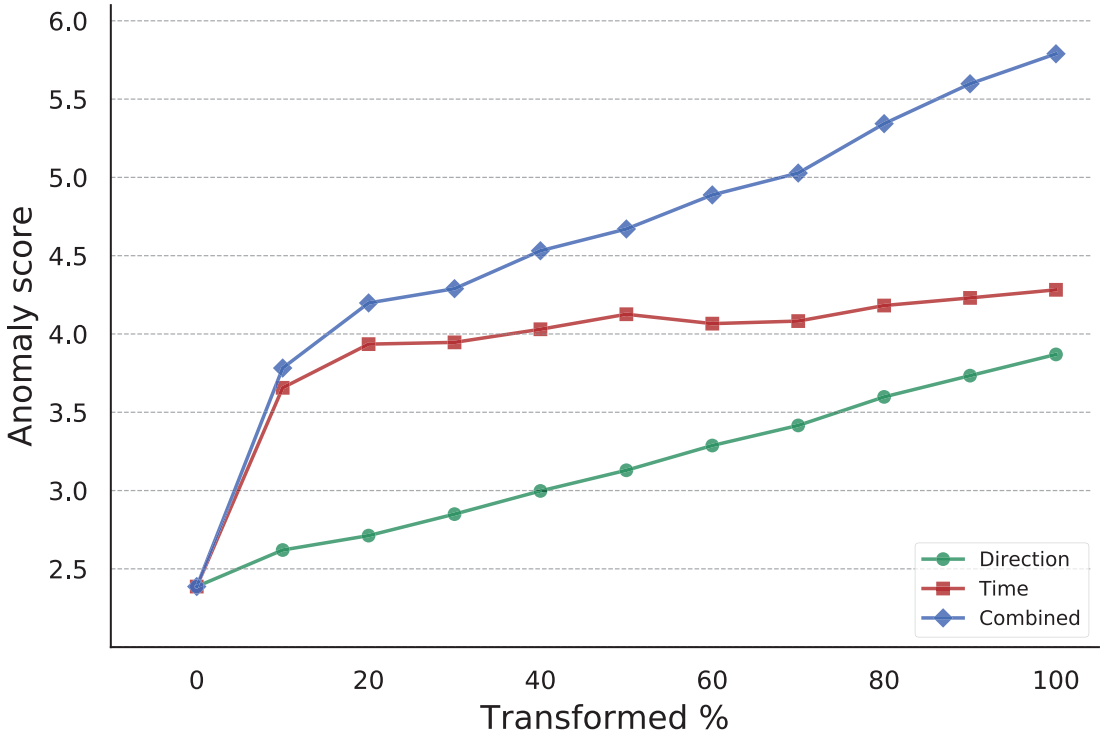


Fig. 2: Plot of test results from each anomaly type. The horizontal axis indicates the percent of samples that were altered. As expected, the mean anomaly score on the vertical axis increases with respect to the amount of altered samples in the test data.

to be unrelated to the type of the anomaly under examination. The statistical properties of network data may fluctuate due to multiple factors.

By using synthetic data which contains correlations that are known to be relevant, it is possible to verify whether or not the proposed network structure is capable of detecting them in general. Moreover, the test may show how the classifier reacts to the increase in variance. In an ideal case a classifier should be relatively tolerant to small fluctuations; however, be able to reliably identify the anomalous samples.

Future work includes refining the statistical procedures, as well as increasing the complexity of correlations in test data. Further research will be conducted on how the relationship between increasing variance and data are drawn from different distributions affects the anomaly score, and how this information may be used to refine the structure of the neural network classifier.

Acknowledgment

This research is funded by:

– *Using Artificial Intelligence for Anomaly Based Network Intrusion Detection System* -project of the Scientific Advisory Board for Defence (MATINE)

– *Cyber Security Network of Competence Centres for Europe (CyberSec4Europe)* -project of the Horizon 2020 SU-ICT-03-2018 program

References

1. Castro, E.R.S., Alencar, M.S., Fonseca, I.E.: Probability density functions of the packet length for computer networks with bimodal traffic. International journal of Computer Networks & Communications **5**(3), 17–31 (2013). DOI 10.5121/ijcnc.2013.5302
2. Chiba, Z., Abghour, N., Moussaid, K., Omri, A.E., Rida, M.: A Clever Approach to Develop an Efficient Deep Neural Network Based IDS for Cloud Environments Using a Self-Adaptive Genetic Algorithm. In: 2019 International Conference on Advanced Communication Technologies and Networking (CommNet), pp. 1–9 (2019). DOI 10.1109/COMMNET.2019.8742390
3. Jain, R., Routhier, S.: Packet trains–measurements and a new model for computer network traffic. IEEE Journal on Selected Areas in Communications **4**(6), 986–995 (1986). DOI 10.1109/JSAC.1986.1146410
4. Kokkonen, T., Puuska, S., Alatalo, J., Heilimo, E., Mäkelä, A.: Network anomaly detection based on wavenet. In: O. Galinina, S. Andreev, S. Balandin, Y. Koucheryavy (eds.) Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pp. 424–433. Springer International Publishing, Cham (2019)
5. Masduki, B.W., Ramli, K., Saputra, F.A., Sugiarto, D.: Study on implementation of machine learning methods combination for improving attacks detection accuracy on intrusion detection system (ids). In: 2015 International Conference on Quality in Research (QiR), pp. 56–64 (2015). DOI 10.1109/QiR.2015.7374895
6. Narsingyani, D., Kale, O.: Optimizing false positive in anomaly based intrusion detection using genetic algorithm. In: 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), pp. 72–77 (2015). DOI 10.1109/MITE.2015.7375291
7. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: Wavenet: A generative model for raw audio (2016). URL https://arxiv.org/pdf/1609.03499.pdf
8. Puuska, S., Kokkonen, T., Alatalo, J., Heilimo, E.: Anomaly-based network intrusion detection using wavelets and adversarial autoencoders. In: J.L. Lanet, C. Toma (eds.) Innovative Security Solutions for Information Technology and Communications, pp. 234–246. Springer International Publishing, Cham (2019)
9. Salimans, T., Karpathy, A., Chen, X., Kingma, D.P.: Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017 (2017). URL https://openreview.net/references/pdf?id=rJuJ1cP_l

P8

MODEL FOOLING ATTACKS AGAINST MEDICAL IMAGING: A SHORT SURVEY

by

T. Sipola, S. Puuska &T. Kokkonen 2020

Information & Security: An International Journal 46, no. 2 (2020): 215–224

https://doi.org/10.11610/isij.4615

Information & Security

T. Sipola, S. Puuska & T. Kokkonen
vol. 46, no. 2 (2020): 215-224
https://doi.org/10.11610/isij.4615

Published since 1998
ISSN 0861-5160 (print), ISSN 1314-2119 (online)

Research Article

Model Fooling Attacks Against Medical Imaging: A Short Survey

Tuomo Sipola **(✉)**, ***Samir Puuska***, ***Tero Kokkonen***

JAMK University of Applied Sciences, Jyväskylä, Finland,
https://www.jamk.fi/en/Home/

ABSTRACT:

This study aims to find a list of methods to fool artificial neural networks used in medical imaging. We collected a short list of publications related to machine learning model fooling to see if these methods have been used in the medical imaging domain. Specifically, we focused our interest to pathological whole slide images used to study human tissues. While useful, machine learning models such as deep neural networks can be fooled by quite simple attacks involving purposefully engineered images. Such attacks pose a threat to many domains, including the one we focused on since there have been some studies describing such threats.

ARTICLE INFO:

RECEIVED: 08 JUN 2020
REVISED: 31 AUG 2020
ONLINE: 13 SEP 2020

KEYWORDS:

deep learning, artificial neural networks, adversarial images, machine learning, medical imaging

Introduction

Artificial Intelligence (AI) based solutions, especially deep learning based on neural networks, are widely used in the medical domain. For example, AI is used for helping and automatizing cancer diagnosis based on image data. The benefit of this approach is to relieve experts to work on more important tasks while automated systems can inspect images and give initial recommendations.

If an attacker can fool the AI processing, ramifications can be devastating. Such attacks may result in incorrect treatment procedures, causing extreme

✉ Corresponding Author: E-mail: tuomo.sipola@jamk.fi

circumstances with a worst-case scenario of losing human lives. In addition, wrong diagnoses could undermine the public trust in medical professionals. This paper presents a short survey of model fooling attacks against neural networks in the medical domain.

Fooling neural networks is an important subject because machine learning models are widely used in medicine for automating many processes and for helping with diagnosis. For example, Rai et al. proposed a convolutional neural network for healthcare assistant [29] while Rastgar-Jazi and Fernando used neural networks for detecting heart abnormalities from electrocardiogram (ECG) data.[30] Similarly, authors of [31] used neural networks for prediction and prevention of heart attacks from ECG data while Murugesan and Sukanesh used neural networks for detecting brain tumours in electroencephalograms (EEG) signals.[24] Syam and Marapareddy discussed three different scenarios of classification problems, where one is skin lesion (cancer) classification from images.[34] As can be seen, these machine learning solutions are useful for many medical applications.

Effectiveness of neural network based deep learning is based on the used algorithm and learning data. If learning dataset is inadequate or contains incorrect information, results will be inaccurate. Similarly, if there are known weaknesses in the used algorithm, they can be compromised. In that sense, AI components can be attacked and fooled to behave incorrectly. As an example of a weakness, Afifi and Brown explore how white balance of photography impact the performance of deep neural networks,[2] while authors of [21] generated adversarial noise for fooling the neural networks. Gu et al. discussed about gradient shielding method for understanding the vulnerabilities in neural networks.[13] MOEA-APGA is an algorithm for achieving targeted attacks against neural networks,[9] and another similar algorithm is called DeepFool implemented for computing perturbations that fool neural networks.[23] In a medical domain, Chuquicusma et.al. studied about fooling radiologists for lung cancer diagnosis.[8] As can be seen, many such attack vectors exist.

As a powerful machine learning method, deep learning has also been applied to images related to pathology, for example, trying to classify images of cancer whole slide images (WSI). Serag et al. present an overview of the application of artificial intelligence for pathology and tissue analytics.[32] As another example, convolutional neural networks have been used for nuclear segmentation, which is an important part of tissue cancer grading.[18] Deliberately produced wrong segmentation could result in wrong diagnoses. Pre-trained convolutional neural networks have been compared to training from scratch using the Kimia Path24 dataset, with results indicating that pre-trained networks are quite competitive.[15] Using such pre-trained models creates a possibility of hidden attacks trained into the model or abusing known deficiencies of such models.

In this study, we collected a list of relevant research papers concerning medical imaging and attacks against neural networks. We queried the publicly available Google Scholar database to identify publications relevant to deep neural network fooling, deep neural networks in medical imaging and deep neural

networks fooled in that domain. The results of this short survey should be useful for anyone trying to understand the vulnerabilities of neural networks in specific domains. Moreover, the use of them in medical imaging raises the question of reliability and robustness when targeted by such attacks. As can be seen, targeted attack against neural networks in medical domain is a realistic scenario. From the attacker perspective, medical domain can be considered as a valuable target because of the critical ramifications of possible attack. In addition, there are known vulnerabilities with neural networks that are highly used in medical domain.

Below, we present the medical imaging domain and then discuss about machine learning regarding that domain. Next sections describe the state of fooling deep neural networks and how it has been applied to the medical domain. Finally, we present our concluding thoughts.

Short Introduction to Whole Slide Images in Cancer Diagnosis

Quick and affordable laboratory cancer diagnosis methods are of great importance. One of the well-established methods is light microscopy with a stain, such as haematoxylin and eosin (H&E). The H&E stain makes various tissue components visible, allowing diagnosis based on e.g. their morphological features.[14]

The advent of digital pathology and whole-slide imaging (WSI) have provided a computerized way to analyse and share the results of light microscopy. By digitizing the tissue images, a variety of automated methods can be used to perform image analysis, annotation, and workflow improvements. Turning glass slides to a digital format requires a slide scanner, which digitizes the slide using specialized format that allows e.g. various zoom levels and metadata to be stored in one data file. This data can then be easily shared, further processed using a variety of tools, and even easily used in teaching in a virtualized microscopy environment.[1]

Distinguishing between benign and malignant tumours is essential for accurate prognosis. One of the features that separate the two is differentiation and anaplasia. In general, benign tumours consist of cells that resemble the tissue where they originated from. They retain much of the functionality and morphology of their non-transformed counterparts but may invade surrounding tissue. Malignant tumours, on the other hand, lose their resemblance to their normal counterparts and become undifferentiated (anaplastic). This change results in noticeable change in cell morphology, and it is possible to observe this using light microscopy and stains. These observable changes include variations in size and shape, nuclear abnormalities and atypical mitoses. Assigning a value to this differentiation is called grading. The criteria and schemes are dependent on the type of tumor.[19]

For breast cancer, observing mitoses has been shown to be a good predictor for tumour development and prognosis. In order to proliferate, tumour cells need to overcome various limitations that prevent ordinary cells from dividing indefinitely. These mutations may result in increased cell-cycle activity, and even majorly affect the mitosis process itself by causing atypical-looking cell

divisions which may be visually observed using light microscopy.[27] Figure 1 shows an example of breast cancer WSI from Al-Janabi et al.[4]

Detecting abnormal morphology and quantifying the number of various cell features is a good candidate for automatization via machine learning and computer vision methods. WSI with sufficient quality can be automatically annotated. Digital pathology is expected to improve convenience and quality of the process. Nam et al. provide an introduction to digital pathology aimed at healthcare professionals.[25] Furthermore, Komura and Ishikawa made a short review of machine learning methods for histopathological image analysis, listing seven whole slide image datasets and 21 hand annotated histopathological datasets.[16]

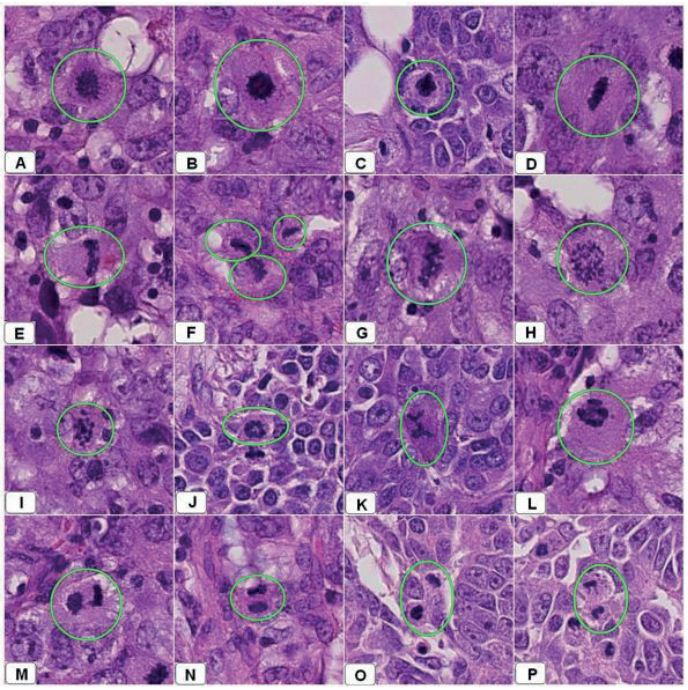


Figure 1: Example of WSI showing several breast resections with infiltrative ductal carcinoma.

Figure courtesy of Al-Janabi et al.,[4] distributed under the terms of the Creative Commons Attribution License.

Fooling Deep Neural Networks

Deep neural networks and deep learning in general refers to a field of study, were complex concepts are learned from simpler representations by creating an interconnected network of activation functions and weights.[11] Due to their nature, these networks may contain flaws which make them susceptible to various classes of errors. These imperfections may be used maliciously to force the network into making an erroneous prediction. There have been several successful attempts at creating methods to fool deep neural networks.

One approach is to give an adversarial image as input to the classifier. Nguyen, Yosinski and Clune used an evolutionary algorithm optimization method to generate unrecognizable images to the human eye. Those images fooled a neural network to classify them as an object with high certainty, even though it should not have. They describe these images as costly exploits that could be used against deep neural networks.[26] Moosavi-Dezfooli, Fawzi and Frossard propose the DeepFool algorithm that efficiently generates adversarial images and quantifies the robustness of image classifiers.[23] An adversarial image is wrongly classified as something else than what the image clearly represents to the human eye. In the paper, a slight perturbation was added to an image of an animal to misclassify a whale as a turtle. Furthermore, it is possible to create adversarial 2D images robust to noise, distortion and affine transformations, and even adversarial 3D printed objects (a turtle).[6]

Adversarial patches are images that can be placed inside another image to fool a neural network classifier. Brown et al. have shown the effectiveness of such images.[7] It is easy to see that inserting such patches to medical images could yield similar results, resulting in a false classification.

Research has already addressed cases of changing only one pixel of an image to cause it to be classified as another object.[33] It is remarkable that a change of colour in one pixel could fool the neural network. A move towards a more theoretical understanding of one pixel-attacks and incorrect mapping to low dimensional manifold has also been proposed. This makes it easy to find localized areas where one-pixel attacks should be more effective.[17]

Backdoored images can be created when attacking the learning stage of a neural network. These malign models can be deployed to production, and the fault is only revealed when the bad image is given as an input, resulting in wrong classification. Outsourced training opens the possibility of creating backdoored neural networks that behave badly on input specified by the attacker.[12] In a similar scheme, called poisoning attack, artificially poisoned data being sent to a model gradually change the model to conform to the attacker's goals. Yang et al. used an autoencoder (instead of the more traditional direct gradient method) to generate poisoned input data for deep neural networks.[37]

The evident vulnerability of neural networks against several types of attacks is alarming because these methods are being proposed in several real-world domains. See [3] for a survey of adversarial attacks against deep learning in computer vision. The authors not only list several attacks but also include defences. They conclude that there is a threat against safety and security critical applications.

Figure 2 shows a schematic presentation of the possible attack routes described above. In this paper we have identified two parts of the AI process, which could be targeted.

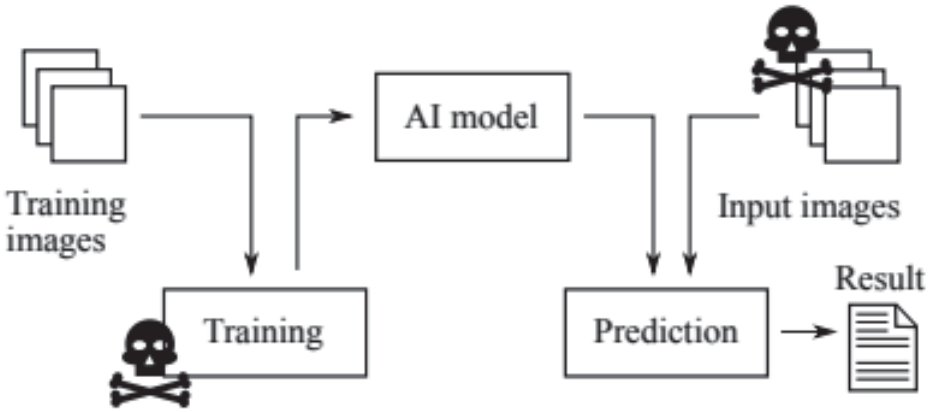


Figure 2: A schematic picture of AI model fooling.
The input images could be altered using adversarial images, patches or one-pixel attacks. In addition, the training process itself could be tampered with.

Fooling Deep Learning in Medical Imaging and Pathology

Although a rather new concern, the vulnerabilities intrinsic to neural network solutions have been identified by the medical community. For evident health reasons the accuracy and robustness of methodology in the medical domain is very important. Tizhoosh and Pantanowitz list challenges and opportunities related to artificial intelligence and digital pathology. One of the challenges concerns adversarial attacks and the shakiness of deep decisions made by neural networks.[36] This fundamental lack of robustness could be one avenue of future research.

Adversarial examples in medical imaging can change the behaviour of classifiers and segmentation, illustrating the lack of robustness in the neural network models. Such approach can also be used for model evaluation.[28] Vulnerability during segmentation could lead to wrong representation of reality during the following stages of diagnosis. Again, the less understood and erratic boundaries of classification are a concern that enable an attack vector.

Deep learning networks classifying X-ray images are also vulnerable to attacks.[35] Being perhaps the most familiar scenario to the public, X-ray image processing is a natural target for automation. However, these kinds of perturbation attacks show that the models can be fooled.

Finlayson et al. successfully use adversarial attacks against medical imaging in three domains: fundoscopy, chest X-ray images and dermoscopy. They also present a risk model for the machine learning pipeline.[10] Patch attacks and projected gradient descent both seem to work against real world images, reducing the reliability of the classifier. However, neural networks can be made robust against perturbation attacks by exploiting the structure of the optimization task.[20]

Not all uses of these methods are harmful. It is possible to use existing medical imaging data to generate more training data and to tackle uneven class balance using various methods, including generative adversarial networks.[22] The methods described above can also be used for beneficial inpainting of missing areas in biomedical imaging. Armanious et al. used generative adversarial networks to inpaint missing areas or incomplete medical images.[5]

The identified fooling methods are listed in Table 1. As can be seen, some of the fooling methods have been used in the medical domain. It should be noted that training process tampering is probably more difficult to execute in practice.

Table 1. Fooling methods against deep neural networks and those in the medical domain.

Method	References	Medical domain
Adversarial images	[26],[23],[6]	[28],[35],[10],[22],[5]
Adversarial patches	[7]	[10]
One-pixel attack	[33],[17]	
Training process tampering	[12],[37]	

Conclusions

Although modern neural networks have proven useful for detecting cancerous cell growth, it is possible to mislead these algorithms. There have been research exploits against deep learning methods, even in the field of pathology. Such exploits include specifically engineered adversarial images, adversarial patches put on actual images, one-pixel attacks and attacks focusing on fooling the training process. The scientific studies this short survey inspected include all those attacks. Even medical imaging is not safe from them, which promotes further study of the underlying causes and robustness problems stemming from the structure of neural networks. The expert opinions from the medical community will also broaden the understanding of the effect of these types of attacks.

Acknowledgements

This research is partially funded by the Cyber Security Network of Competence Centres for Europe (CyberSec4Europe) project of the Horizon 2020 SU-ICT-03-2018 program.

References

[1] Famke Aeffner, Mark Zarella, Nathan Buchbinder, Marilyn M Bui, Matthew Goodman, Douglas Hartman, Giovanni Lujan, Mariam Molani, Anil Parwani, Kate Lillard, Oliver Turner, Venkata Vemur, Ana Yuil-Valdes, and Douglas Bowman, "Introduction to Digital Image Analysis in Whole-slide Imaging: A White Paper from the Digital Pathology Association," *Journal of pathology informatics* 10, no. 1 (2019): 9.

[2] Mahmoud Afifi and Michael S Brown, "What Else Can Fool Deep Learning? Addressing Color Constancy Errors on Deep Neural Network Performance," In: *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 243–252.

[3.] Naveed Akhtar and Ajmal Mian, "Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey," *IEEE Access* 6 (2018): 14410–14430.

[4] Shaimaa Al-Janabi, Henk-Jan van Slooten, Mike Visser, Tjeerd van der Ploeg, Paul J. van Diest, and Mehdi Jiwa, "Evaluation of Mitotic Activity Index in Breast Cancer Using Whole Slide Digital Images," PloS one 8, no. 12 (2013).

[5] Karim Armanious, Youssef Mecky, Sergios Gatidis, and Bin Yang, "Adversarial Inpainting of Medical Image Modalities," In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE*, 2019, pp. 3267–3271.

[6] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok, "Synthesizing Robust Adversarial Examples," In: *Proceedings of the 35th International Conference on Machine Learning, PMLR, Stockholm, Sweden, vol. 80,* 2018, pp. 284–293.

[7] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer, "Adversarial Patch," arXiv e-prints (2018), https://arxiv.org/abs/1712.09665.

[8] Maria Chuquicusma, Sarfaraz Hussein, Jeremy Burt, and Ulas Bagci, "How to Fool Radiologists with Generative Adversarial Networks? A Visual Turing Test for Lung Cancer Diagnosis," In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), IEEE,* 2018, pp. 240–244.

[9] Yepeng Deng, Chunkai Zhang, and Xuan Wang, "A Multi-objective Examples Generation Approach to Fool the Deep Neural Networks in the Black-box Scenario," In: *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), IEEE,* 2019, pp. 92–99.

[10] Samuel Finlayson, Hyung Won Chung, Isaac Kohane, and Andrew Beam, "Adversarial Attacks Against Medical Deep Learning Systems," *arXiv e-print* (4 Feb. 2019), https://arxiv.org/abs/1804.05296.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning* (MIT Press, 2016), http://www.deeplearningbook.org.

[12] Tianyu Gu,Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg, "Badnets: Evaluating Backdooring Attacks on Deep Neural Networks," *IEEE Access* 7 (2019): 47230–47244.

[13] Zhaoquan Gu, Weixiong Hu, Chuanjing Zhang, Hui Lu, Lihua Yin, and Le Wang, "Gradient Shielding: Towards Understanding Vulnerability of Deep Neural Networks," *IEEE Transactions on Network Science and Engineering,* 2020.

[14] Luis Carlos Junqueira and Jose Carneiro, *Basic Histology Text & Atlas* (McGraw-Hill Professional, 2005).

[15] Brady Kieffer, Morteza Babaie, Shivam Kalra, and H.R.Tizhoosh, "Convolutional Neural Networks for Histopathology Image Classification: Training vs. Using Pre-trained Networks," In: *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), IEEE,* 2017, pp. 1–6.

[16] Daisuke Komura, Shumpei Ishikawa, "Machine Learning Methods for Histopathological Image Analysis," *Computational and structural biotechnology journal* 16 (2018): 34–42.

[17] David Kügler, Alexander Distergoft, Arjan Kuijper, and Anirban Mukhopadhyay, "Exploring Adversarial Examples," In: *Understanding and Interpreting Machine Learning in Medical Image Computing Applications* (Springer, 2018), 70–78.

[18] Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi "A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology," *IEEE Transactions on Medical Imaging* 36, no. 7 (2017): 1550–1560.

[19] Vinay Kumar, Abul Abbas, and Jon Aster, *Robbins Basic Pathology*, 10th edition (Saunders, Philadelphia: Elsevier, 2017).

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," *arXiv e-prints,* arXiv:1706.06083 (2019).

[21] Marko Mihajlović and Nikola Popović, "Fooling a Neural Network with Common Adversarial Noise," In: *2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON), IEEE*, 2018, pp. 293–296.

[22] Agnieszka Mikołajczyk and Michał Grochowski, "Data Augmentation for Improving Deep Learning in Image Classification Problem," In: *2018 international interdisciplinary PhD workshop (IIPhDW), IEEE*, 2018, pp. 117–122.

[23] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: A Simple and Accurate Method to Fool Deep Neural Networks," In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE,* 2016, pp. 2574–2582.

[24] Muthukumar Murugesan and R. Sukanesh, "Automated Detection of Brain Tumor in EEG Signals Using Artificial Neural Networks," In: *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, IEEE*, 2009, pp. 284–288.

[25] Soojeong Nam, Yosep Chong, Chan Kwon Jung, Tae-Yeong Kwak, Ji Youl Lee, Jihwan Park, Mi Jung Rho, and Heounjeong Go, "Introduction to Digital Pathology and Computer-aided Pathology," *The Korean Journal of Pathology* 54, no. 2 (2020): 125-134.

[26] Anh Nguyen, Jason Yosinski, and Jeff Clune, "Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images," In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE,* 2015, pp. 427–436.

[27] Ryuji Ohashi, Shigeki Namimatsu, Takashi Sakatani, Zenya Naito, Hiroyuki Takei, and Akira Shimizu, "Prognostic Utility of Atypical Mitoses in Patients with Breast Cancer: A Comparative Study with ki67 and Phosphohistone h3," *Journal of surgical oncology* 118, no. 3 (2018): 557–567.

[28] Magdalini Paschali, Sailesh Conjeti, Fernando Navarro, and Nassir Navab, "Generalizability vs. Robustness: Adversarial Examples for Medical Imaging," *arXiv e-prints* (2018).

[29] Siddhant Rai, Akshayanand Raut, Akash Savaliya, and Radha Shankarmani, "Darwin: Convolutional Neural Network based Intelligent Health Assistant," In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), IEEE,* 2018, pp. 1367–1371.

[30] Maryam Rastgar-Jazi and Xavier Fernando, "Detection of Heart Abnormalities via Artificial Neural Network: An Application of Self Learning Algorithms," In: *2017 IEEE Canada International Humanitarian Technology Conference (IHTC), IEEE*, 2017, pp. 66–69.

[31] Deepika Ravish, Nayana R Shenoy, K. Shanthi, and S. Nisargh, "Heart Function Monitoring, Prediction and Prevention of Heart Attacks: Using Artificial Neural Networks," In: *2014 International Conference on Contemporary Computing and Informatics (IC3I), IEEE,* 2014, pp. 1–6.

[32] Ahmed Serag, Adrian Ion-Margineanu, Hammad Qureshi, Ryan McMillan, Marie-Judith Saint Martin, Jim Diamond, Paul O'Reilly, and Peter Hamilton, "Translational AI and Deep Learning in Diagnostic Pathology," *Frontiers in Medicine* 6 (2019): 185.

[33] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi, "One Pixel Attack for Fooling Deep Neural Networks," *IEEE Transactions on Evolutionary Computation* 23, no. 5 (2019): 828–841.

[34] Rajarshi Syam and R. Marapareddy, "Application of Deep Neural Networks in the Field of Information Security and Healthcare," In: *2019 SoutheastCon, IEEE*, 2019, pp. 1–5.

[35] Saeid Asgari Taghanaki, Arkadeep Das, and Ghassan Hamarneh. "Vulnerability Analysis of Chest X-ray Image Classification against Adversarial Attacks," In: *Understanding and Interpreting Machine Learning in Medical Image Computing Applications* (Springer, 2018): 87–94.

[36] Hamid Reza Tizhoosh and Liron Pantanowitz, "Artificial Intelligence and Digital Pathology: Challenges and Opportunities," *Journal of Pathology Informatics* 9 (2018).

[37] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen, "Generative Poisoning Attack Method against Neural Networks," *arXiv e-prints* (2017).

P9

ONE-PIXEL ATTACK DECEIVES COMPUTER-ASSISTED DIAGNOSIS OF CANCER

by

J. Korpihalkola, T. Sipola, S. Puuska &T. Kokkonen 2020

Computers & Security (under review)

https://arxiv.org/abs/2012.00517

One-pixel attack deceives automatic detection of breast cancer

Joni Korpihalkola[a], Tuomo Sipola[a,*], Samir Puuska[b], Tero Kokkonen[a]

[a]*Institute of Information Technology, JAMK University of Applied Sciences, PO Box 207, FI-40101 Jyväskylä, Finland*
[b]*Faculty of Information Technology, University of Jyväskylä, PO Box 35, FI-40014 Jyväskylä, Finland*

Abstract

In this article we demonstrate that a state-of-the-art machine learning model predicting whether a whole slide image contains mitosis can be fooled by changing just a single pixel in the input image. Computer vision and machine learning can be used to automate various tasks in cancer diagnostic and detection. If an attacker can manipulate the automated processing, the results can be devastating and in the worst case lead to wrong diagnostic and treatments. In this research one-pixel attack is demonstrated in a real-life scenario with a real tumor dataset. The results indicate that a minor one-pixel modification of a whole slide image under analysis can affect the diagnosis. The attack poses a threat from the cyber security perspective: the one-pixel method can be used as an attack vector by a motivated attacker.

Keywords: adversarial examples, cyber security, machine learning, medical imaging, breast cancer, model safety

1. Introduction

Cancer, in its various forms, is one of the leading causes of death in the western world. A number of detected cancers of a determined type in a defined population during a year is expressed as cancer incidence rate (CIR), commonly formed as the number of

*Corresponding author: Tel.: +358 50 310 3339;
Email addresses: joni.korpihalkola@jamk.fi (Joni Korpihalkola), tuomo.sipola@jamk.fi (Tuomo Sipola), sapepuus@student.jyu.fi (Samir Puuska), tero.kokkonen@jamk.fi (Tero Kokkonen)

cancers per 100,000 population (U.S. National Cancer Institute at the National Institutes of Health (NIH), a). According to the U.S. National Cancer Institute at the National Institutes of Health, the CIR, based on 2013–2017 statistics in the U.S., is 442.4 per 100,000 men and women per year (U.S. National Cancer Institute at the National Institutes of Health (NIH), b). The institute stated that *"Cancer is among the leading causes of death worldwide. In 2018, there were 18.1 million new cases. The number of new cancer cases per year is expected to rise to 29.5 million"* (U.S. National Cancer Institute at the National Institutes of Health (NIH), b).

The high number of new incidences (high CIR) means that cancer and various cancer-related medical tasks require substantial time and resources. Although cancer is not one disease, many laboratory diagnostic methods are shared between various types. These include morphologic methods where microscopy in combination with various staining methods is used to draw conclusions based on various properties and counts of cells in a biological sample. In many common types of cancers, early detection is a key factor in improving the prognosis (van Diest et al., 2004). Since detection plays a major role in patient outcomes, automating some of this work using techniques such as machine learning can lead to faster detection, increased throughput, and reduced costs.

From the cyber security standpoint this increased automation means increased attack surface. Disrupting the operation of nation's critical infrastructure has been an integral part of nation-state level attacks. The goal of these attacks may simply be to erode trust in the nation's capability to provide services to citizens, or in the worst case, be a part of an armed conflict. Healthcare sector is a major part of critical infrastructure, and as such a target for advanced cyber operations. Although cyber operations against computers, networks, and data required for administering medical care are prohibited under international law, there has been a steady increase in attacks against them (Schmitt, 2017). Awareness has an important role in cyber security of the healthcare sector, as stated by Rajamäki et al. (2018): *"The highest concern for healthcare organizations is the employee negligence followed by the fear of a cyber-attack."*

Automated or partly automatized analysis systems are priority targets for cyber attacks, as a disruption in these systems causes major effects resulting in decreased capability to diagnose and treat patients, increased expenses, and overall reduction in

trust towards automated systems. In their study, Spanakis et al. analyzed cyber security in the healthcare domain and stated the fact that growth of technology utilized in healthcare concurrently increases the attack surface and thus the risk of cyber incidents increases (Spanakis et al., 2020).

In cancer diagnosis, computer vision and machine learning can be used to automate various tasks (Zhang et al., 2017; Nasief et al., 2019) where e.g. a count of cells needs to be made from an image (Veta et al., 2016). Although various approaches for analyzing shapes have been proposed in the literature, one of the newer approaches is to use an artificial neural network for detecting the desired properties. These tools aim to make cancer diagnosis less expensive and less time consuming (Khosravi et al., 2018; Bera et al., 2019; Alom et al., 2019).

Papernot et al. state that models of machine learning are vulnerable to modified (malicious) inputs and on that account they introduced a black-box attack against deep neural networks without knowledge of the classifier training data or model (Papernot et al., 2017). Such attack methods have been introduced in some real world scenarios, for example, Stokes et al. (2018) studied the attack, and furthermore, defence of malware detection models and image modifications against artificial intelligence (AI) based computer vision capabilities has been researched in (Kang et al., 2020). When attacking against computer vision and image based machine learning, pixel modification is an evident possibility. Lin et al. tested adversarial attacks by modifying critical pixels of the image with limitations for the number of modified pixels (Lin et al., 2020). One-pixel attack is a more advanced method, in which only one pixel of an image is modified in order to fool the classifier (Su et al., 2019). Additionally, mitigation capabilities have been developed, Paul et al. (2020) introduce mitigation of adversarial attacks on medical image systems with the conclusion that its effectiveness can be decreased by adding adversarial images in the training set. In addition to this kind of robust optimization, Xu et al. mention the possibility of gradient masking and attack detection before forwarding the images to the actual classifier. (Xu et al., 2020)

The healthcare sector can be seen as a valuable target for cyber attackers with different motivations. One possible motivation can be the capability of claiming ransoms. Modifying the automated diagnosis capability with a cyber attack may affect the treat-

ment and in the worst case scenario lead to loss of human lives. That also raises the possibility of targeted attacks against a particular person. In conclusion, such attacks may lead to global lack of trust in automated diagnosis systems (Sipola et al., 2020).

In this study we attempt to construct images that fool real world state-of-the-art classifiers. In addition, we explore how to create fake images that appear authentic also to the human observer. Thirdly, we present a method for altering existing images to achieve the goal. We opted to use full slide microscopy images, as they are a major target for automated analysis and digital pathology research, as well as being readily available for scientific use. Although the proposed methods are presented in a cyber attack context, the results can be used to improve or assess classifiers outside this context.

2. Methods and experimental setup

Morphological methods leverage various features of cells in deciding whether they have neoplastic characteristics. The cells are extracted from the tissue under study, and visually inspected using microscopy and stains. The so called whole slide images are high resolution pictures that are often digitized and viewed using computers. This format also provides a suitable basis for a more automated approach.

In this article we examine classifying methods based on artificial neural networks. An artificial neural network classifiers are usually trained using samples, i.e. images, that have a known label in a fashion that allows the trained classifier to recognize also samples that it has not previously "seen".

Goodfellow et al. define adversarial examples as samples where an adversary makes a small but well-chosen perturbation into an input sample causing the artificial neural network classifier to misclassify that sample with high confidence (Goodfellow et al., 2015). If an adversary possesses a fast way of generating the adversarial examples, they can be used to mount various types of attacks against systems that utilize neural networks for classification tasks. In this article we show that creating adversarial examples in a context of medical imaging is both feasible and fast. Furthermore, we show that a particular type of perturbation is sufficient to alter the vast majority of le-

gitimate input images in a fashion that causes the classifier to misclassify them with high confidence.

Two attacks were performed: mitosis-to-normal, where the objective is to minimize the confidence score value, and normal-to-mitosis, where the objective is to maximize the confidence score value. The former attack type alters an image containing abnormal mitosis into one where the classifier fails to detect this with high confidence. The latter converts an image with normal mitosis into one where the classifier misclassifies it being an abnormal mitosis.

2.1. Attack target

The dataset used in this research is from the Tumor Proliferation Assessment Challenge 2016 (TUPAC16) (Medical Image Analysis Group Eindhoven (IMAG/e), 2016; Veta et al., 2019). The dataset consists of 500 whole slide light microscopy images with known tumor proliferation scores, ground truth labels for the training set, as well as region of interest location data for 148 images.

The dataset was preprocessed by a script from IBM CODAIT Center for Open-source Data & AI Technologies' *deep-histopath* repository,[1] which split the whole slide image into 64-by-64 pixel PNG-format images. The images were marked either 'mitosis' or 'normal' according to the provided labeling.

The chosen classifier was IBM CODAIT's MAX breast cancer detector (Dusenberry and Hu, 2018). This classifier was chosen for its high ranking in the TUPAC16 challenge, and the open source nature of the code. Due to the nature of artificial neural network -based classifiers, this attack method is likely to work on other TUPAC16 contest entries as well. The obtained results do not suggest a particular failure or error in the IBM CODAIT's work or approach.

To simulate a black-box attack situation, the artificial neural network is queried through a HTTP API. Only the input image and the confidence score of the artificial neural network model for the input image are known. Inference on the model was performed by converting the image to a byte string and querying the model API residing

[1] `https://github.com/CODAIT/deep-histopath`

in a Docker container. The response from the API returned a confidence score for the image. The images were also filtered based on the confidence score provided by the artificial neural network. A 'mitosis' labeled image with confidence score below 0.9 and 'normal' labeled image with score above 0.1 are filtered out of the experiment. This way the attacks focus on the unambiguous cases that should be classified correctly by the artificial neural network. Computation time was capped at five days. Consequently, 5,343 'mitosis' and 80,725 'normal' labeled images are tested using this method.

2.2. Attack outline

The goal of the attack is to find a method capable of perturbing legitimate input images in a way that causes the classifier to misclassify them with high confidence. It is usually in the interest of the attacker to find a perturbation that alters the original image as little as possible. The so-called *one-pixel attack* is achieved when the perturbation that causes a misclassification consists of altering just one pixel in the input image. To a human observer the difference between the original and altered image might be indistinguishable. As stated, two attacks are performed: mitosis-to-normal, where the objective is to minimize the confidence score value and normal-to-mitosis, where the objective is to maximize the confidence score value.

To carry out a black-box attack the adversary needs to make perturbations to the original image, and observe how the classifier under attack reacts. Su et al. proposed a method capable of creating one-pixel perturbations using differential evolution (Su et al., 2019). Differential evolution is an optimization method (Feoktistov, 2006; Price, 2013) which can be leveraged for iteratively refining the chosen perturbations until the attacker achieves the desired misclassification confidence. In this study, we used the implementation of differential evolution in the Scikit-learn library (Pedregosa et al., 2011).

A color digital image can be presented as a grid of pixels, where each pixel is a mix of red, green, and blue colors, corresponding to the color sensing cells in human eye. A one-pixel perturbation can be represented by a vector: $\mathbf{x} = (x, y, r, g, b)$, where x and y are the pixel coordinates and r, g, b are the red, green and blue values of the color. All these variables are integers. The bounds for coordinates are [0,63] and the

bounds for color values are [0,255].

The initial population consists of 200 one-pixel perturbation attack vectors, the vector values are initialized using Latin hypercube sampling, which ensures that each coordinate and color value is uniformly sampled inside its bounds. A larger initial population was found to increase attack success only in some rare cases, while it slowed down attack vector search considerably due to higher computation costs. The mutation factor was set at 0.5 and the recombination factor at 0.7. Larger mutation factor and lower recombination factor values were not found to impact the attack success rate in neither mitosis-to-normal nor normal-to-mitosis attacks. Maximum iterations for the evolution were set at 100, although in practice the evolution converged on average at 44 iterations in mitosis-to-normal and 39 on normal-to-mitosis attacks.

After the initial population is created, the members of the population are iterated over. The strategy for creating trial vectors was chosen as 'best1bin'. In the strategy, two random vectors are chosen to mutate the best performing vector in the population, meaning the attack vector that achieved the lowest value from the artificial neural network output. The parameters from the best vector are mutated using the mutation factor and the difference of the two random vectors.

A trial vector is created. Random values for each parameter are generated using binomial distribution; if the random value for the parameter is lower than the recombination factor value, the mutated value in the parameter is inserted into the trial vector. If the random value is higher than the recombination factor value, the value from the best performing vector is carried over into the trial vector. However, one random value is always replaced with a mutated value, even if the binomial distribution values were all below the recombination factor value.

After the trial vector is created, its performance is tested by modifying the target image with the vector's values. If the trial vector performs better than the original member of the population, it is replaced with the trial vector. If the trial member also performs better than the best member of the population, the best member is also replaced by the trial vector.

When the attack vector population is iterated over, a convergence check is performed. The convergence check compares if the standard deviation of the population's

confidence scores is lower than the absolute mean of the population's confidence scores multiplied by tolerance factor and the absolute tolerance value is added to the result multiplication result. If the comparison is true, the algorithm converges and the best population member is the best attack vector found and its confidence score is the function's minimum value.

2.3. Attack success metric

The first criterion for a successful attack is the number of steps in its evolution progress. Attacks that converged after the iteration of the initial population were found to not alter the confidence score at all or by very little margin. Thus, a successful attack needs to iterate the population more than once.

The closer the model's confidence score is to 1, the more sure the model is that the image should be labeled 'mitosis' and the closer the score is to 0, the image is to be labeled 'normal'. To define attacks as successful, mitosis and normal attacks should reach at least 0.5 score threshold, reducing the neural network's prediction into a coin flip. If a mitosis-to-normal attack manages to lower confidence score to 0.05 or a normal-to-mitosis attack the score to 0.95, the model is fooled to predict the opposite label with high certainty.

3. Results

The one-pixel attack was performed on 5,343 'mitosis' labeled images and 80,725 'normal' labeled images. The attack results were documented in a comma-separated values (CSV) file, including the name of the images used in the experiment, differential evolution parameters, original confidence score and the score after the attack. The confidence score indicates how confident the artificial neural network is that the image contains mitosis activity. The score varies between [0, 1], where 0 means that the image is considered normal and 1 means that it is considered to contain mitoses.

3.1. Failures due to early convergence

Attacks where evolution converged immediately after the initial population can be considered as failed attacks. In mitosis-to-normal attacks, in 1,594 or approximately 30%

of the attacks the algorithm converged already after the calculation of initial population function values, while in normal-to-mitosis attacks, 80,520 or 99.7% converged after the initial population. This is due to the tolerance value set at 0.01 and the standard deviation of the initial population being too low compared to tolerance value multiplied by mean of initial population. Lowering of the tolerance value had no impact on finding more successful populations. The tolerance value and the convergence check cause the amount of 'normal' labeled images processed to be higher than 'mitosis' labeled, because more evolution steps were performed on 'mitosis' labeled images. If the evolution converges after the initial population values, in mitosis-to-normal attacks the attack yields only 0.06 change in confidence score on mean and in normal-to-mitosis attacks the confidence score change is 0.001 on mean.

3.2. Confidence scores

The changes in the confidence score were noticeable in both attack types. On mitosis-to-normal attack, 3,407 attacks (91%) out of 3,749 managed to lower the artificial neural network's confidence score below 0.5 and 895 attacks (24%) lowered it below 0.05. On normal-to-mitosis attacks, neural network's confidence score was raised higher than 0.5 on 173 out of 205 attacks (84%) but none of the attacks managed to cross above the 0.95 score threshold.

When looking at attacks where the differential evolution algorithm did not converge on the initial population, the median confidence score difference between the original score and score after attack reaches 0.81. Applying the same filter as in mitosis-to-normal attacks, the median confidence score difference in normal-to-mitosis attacks between original images and attacked images reaches 0.27.

Mitosis-to-normal attacks were successful in finding adversarial examples. Figure 1 has its center line at the median value, its box limits extending from 25% to 75%, its whiskers from the edges of the box to no more than 1.5 times interquartile range, ending at the farthest point in the interval and its outliers plotted as dots. The figure shows how the neural network's confidence score before the attack is on average 0.96, the maximum score is 0.99 and minimum is 0.90. After attacking the images and finding adversarial images, artificial neural network's confidence score median values

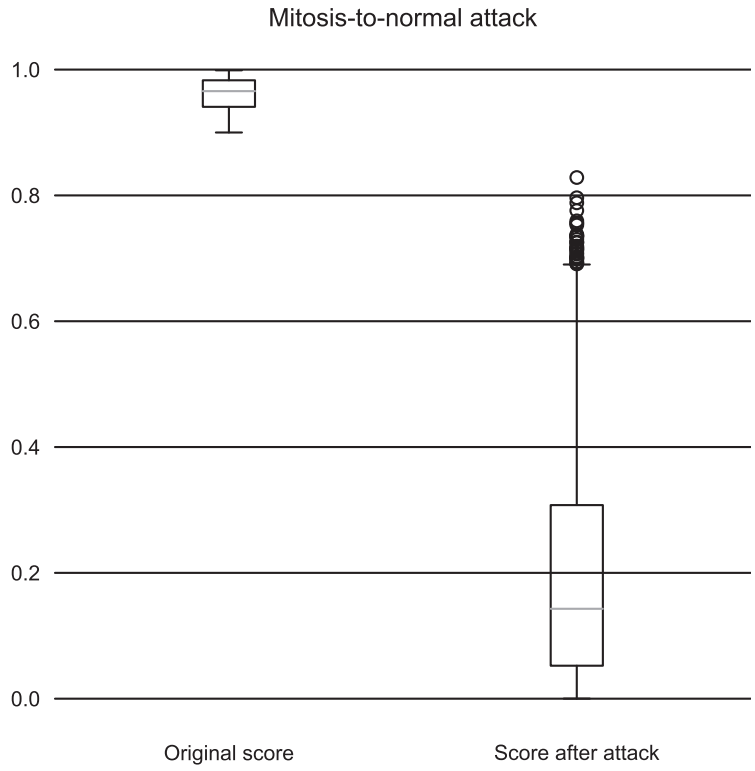


Figure 1: Box plot visualization of mitosis-to-normal attack experiment confidence scores. The majority of the attacks were successful, lowering the confidence score below 0.5 in 3,407 or approximately 91% of the attacks. 895 or approximately 24% of the attacks manage to lower the confidence score below 0.05.

	Before attack	After attack
Maximum	0.99	0.83
Mean	0.96	0.20
Median	0.96	0.14
Standard deviation	0.02	0.18
Minimum	0.99	0.00011

Table 1: Confidence score statistics for mitosis-to-normal attack, where the number of attacks is 3,749.

are 0.1, they also reach a minimum of 0.0001 and a maximum of 0.83. The standard deviation for the scores is 0.18 and the mean is 0.20. This information is also conveyed in Table 1.

Normal-to-mitosis attacks were also successful. Before the images are attacked, neural network's confidence scores are on average 0.048, where the minimum is 0.0036

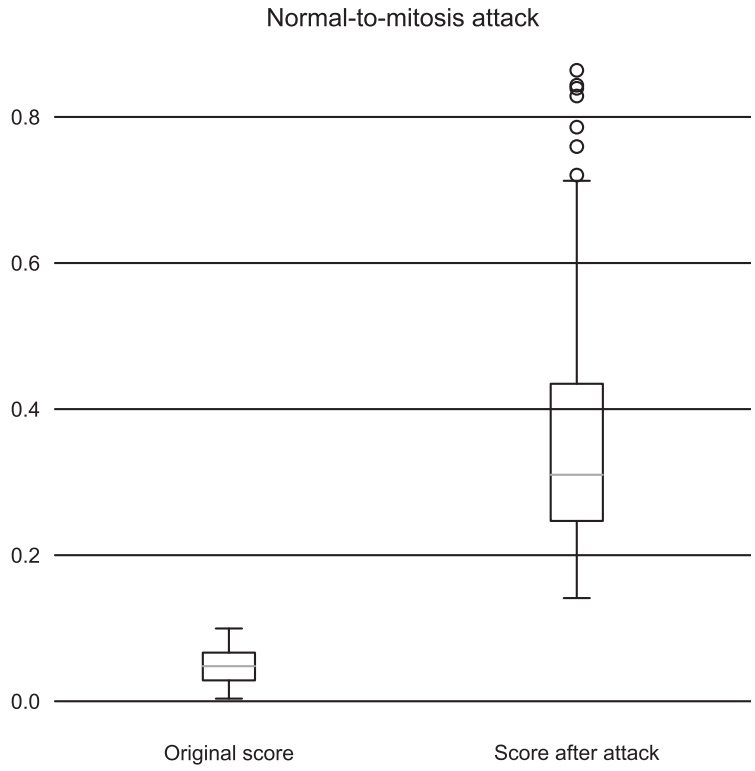


Figure 2: Box plot visualization of normal-to-mitosis attack experiment confidence scores, where 173 or approximately 84% of attacks manage to raise the artificial neural network's confidence score above 0.5. None of the attacks manage to cross above the 0.95 score threshold.

	Before attack	After attack
Maximum	0.099	0.86
Mean	0.048	0.36
Median	0.048	0.31
Standard deviation	0.025	0.15
Minimum	0.0036	0.14

Table 2: Confidence score statistics for normal-to-mitosis attack, where the number of attacks is 205.

and maximum is 0.099. Figure 2 shows this as box plot, which shares its statistical characteristics with 1. After attacking the images, neural network's confidence score median is 0.31, the scores minimum reaches 0.14 and maximum 0.86. The standard deviation for the scores is 0.15 and the mean is 0.36. This information is also conveyed in Table 2.

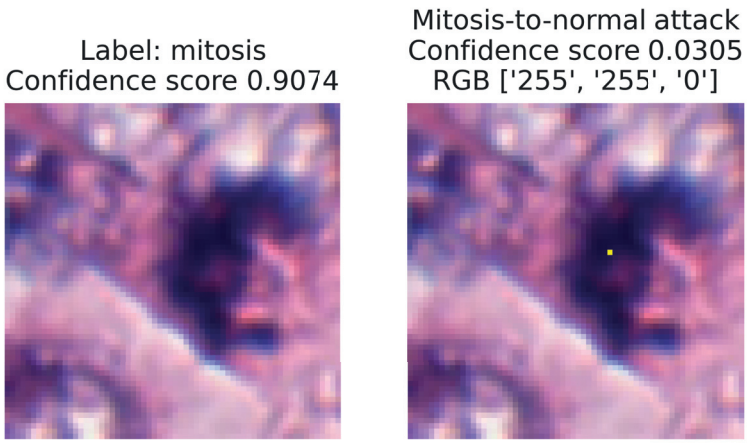


Figure 3: An adversarial example that is misclassified as normal even though in reality the source image is labeled as having mitosis activity. Notice the bright yellow pixel inside the dark area in the middle right part of the image.

3.3. Adversarial examples

As an example, we showcase two successful attacks. Firstly, Figure 3 shows this adversarial example that deceives the predictor to think that an image containing mitosis is a normal picture without any signs of disease. In the attacks, the most common pixel color was pure yellow, meaning RGB values (255, 255, 0), which was used in 2,214 attacks. In 122 attacks the pixel color was pure white, meaning RGB values (255, 255, 255), which was used in 122 attacks. In the rest of the attacks the pixel colors were yellow with a slightly higher blue value. Secondly, Figure 4 shows an adversarial example that deceives the predictor to think that a picture of normal cell activity contains mitosis. The most common pixel color RGB values was pure yellow (255, 255, 0) and the second most common was pure white (255, 255, 255) and the third was pure black (0, 0, 0). There is a larger variety of colors in attack vectors than in mitosis-to-normal-attacks, but this is most likely explained due to the low amount of successful attacks.

We provide the evolutionary convergence plots for both of the images. Figure 5 shows the progress of differential evolution for the attacked image shown in Figure 3. The lowest confidence score already reaches to almost 0.5 during the initial population attacks and drops down below 0.1 in a few steps. The minimum is reached after 40

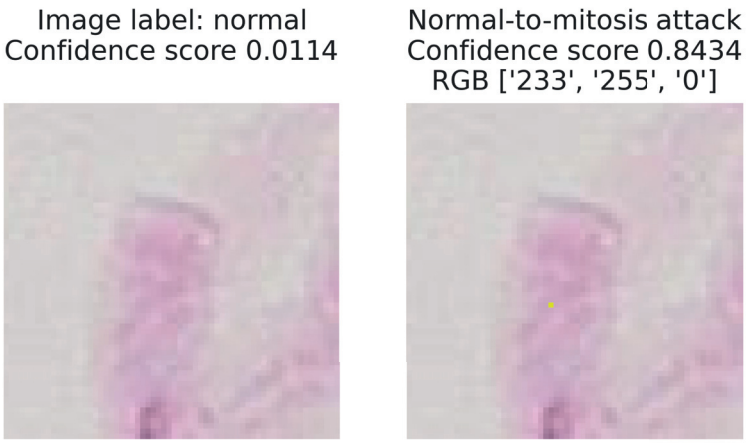


Figure 4: An adversarial example that is misclassified as mitosis even though in reality the source image is labeled as having no mitosis activity. Notice the yellow-lime dot in the middle of the image.

steps. Figure 6 shows the progress of differential evolution for the image and an adversarial image in Figure 4. The maximum score of the initial population attacks is still quite low, below 0.4, but the maximum score of the population quickly rises to near 0.8 in 10 steps. The maximum 0.84 score is reached after 30 steps of the differential evolution algorithm.

4. Discussion

This research demonstrates that one-pixel attacks are successful against artificial neural network analysis of mitosis images. It shows that a machine learning model can perform acceptably with the training and testing sets but fails catastrophically when an adversarial example is used as input. This hilights the need of ensuring the robustness of these artificial neural network models. While it is evident that the model works as expected in the common case, data reproduction and transmission errors, as well as cyber attacks of only one pixel, could produce undesirable results.

It is evident that the attack against mitosis images is the easier one. These images might be of a more varied nature than the normal tissue images. Because of this, modifying the mitosis images does not create as considerable a change as when modifying normal tissue images. On the other hand, deceiving the artificial neural network with modified normal images was more difficult. We speculate that the neural network

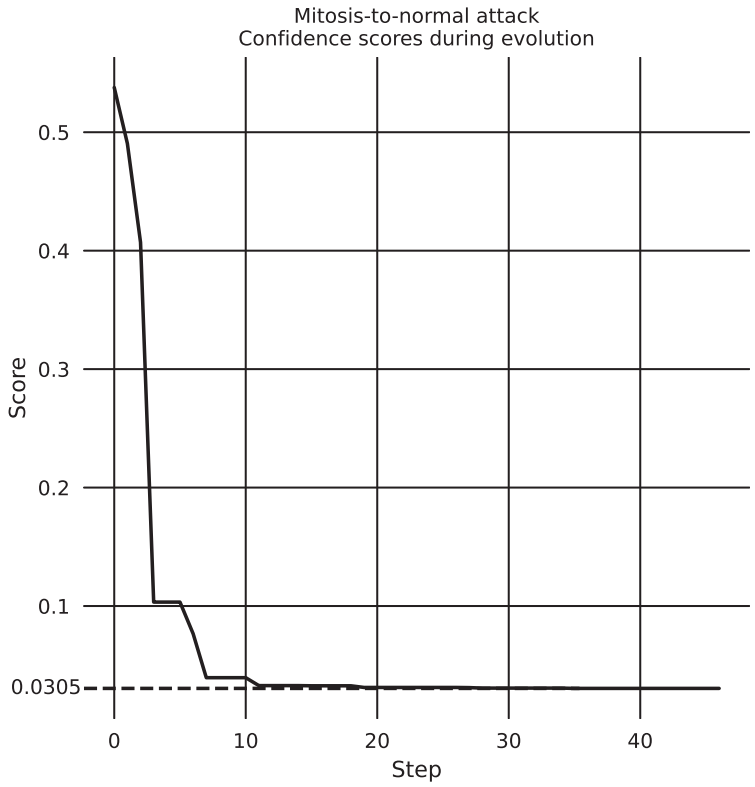


Figure 5: Lowest neural network confidence scores during steps of differential evolution. Example of of an attack against one image, in this case the same as in Figure 3.

has likely learned to classify images with large black blobs as mitosis, thus the neural network is not easily fooled to change labels by only modifying one pixel. Larger modification of the input image would be needed for higher normal-to-mitosis attack success rate.

This result should not be taken as a discouragement of the use of automated diagnosis systems as part of medical imaging. Instead, it shows that the medical models built using modern artificial neural network technologies can be vulnerable to unexpected attacks. As with all technology, its limitations should be known in order to correctly utilize the capabilities it provides.

From the cyber security point of view this should be considered as an alarming finding because motivated and skilled attackers can execute such attacks quite easily, if they have access to the medical image repository. This real-life scenario can be considered possible because there are more and more attacks against healthcare systems

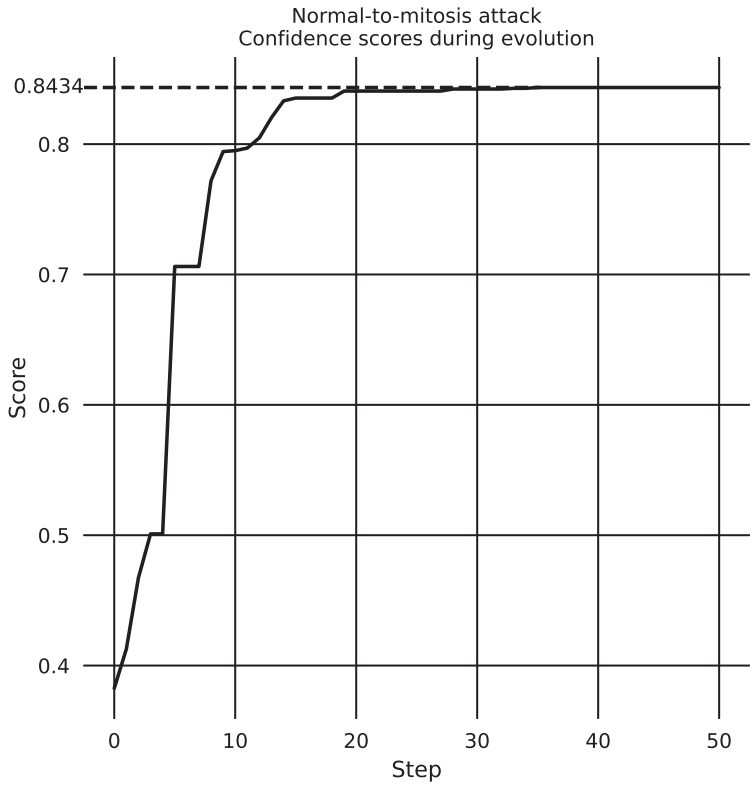


Figure 6: Highest neural network confidence scores during steps of differential evolution. Example of of an attack against one image, in this case the same as in Figure 4.

and with such an attack scenario the motivation can be ransoms. Attackers might think that there is a high probability that ransoms will be paid when a threat of misdiagnosis of cancer exists.

Attack methods will keep evolving. At the moment, the attack pixel may be somewhat prominent, which makes their detection easy, although such artifacts could be introduced just before the analysis. In the future, attack-side research ideas could include blending the attack vector color values as seamlessly to the surrounding pixels as possible, thus fooling human observers.

Data availability

The whole slide images and labeling for the training dataset are available from the Tumor Proliferation Assessment Challenge 2016 (TUPAC16) website (`http://tupac.tue-image.nl/`), which derives the data from the Cancer Genome Atlas by the TCGA

Research Network (https://www.cancer.gov/tcga), the AMIDA13 dataset and from two different pathology centers in The Netherlands (Medical Image Analysis Group Eindhoven (IMAG/e), 2016; Veta et al., 2019; U.S. National Cancer Institute at the National Institutes of Health (NIH), c; Veta et al., 2015). Restrictions apply to the TUPAC16 dataset, as the labels are available for registered users only. However, the data are available upon reasonable request, through collaborative investigations and with permission of Medical Image Analysis Group Eindhoven (IMAG/e). Furthermore, the experiment results are available in CSV format in the same repository as the programming codes used in the experiment, available from the corresponding author upon reasonable request.

Competing interests statement

The authors declare no competing interests.

Author contributions

Joni Korpihalkola: Software, Formal analysis, Writing - Original draft, Visualization. **Tuomo Sipola:** Conceptualization, Methodology, Validation, Data curation, Writing - Original draft, Supervision. **Samir Puuska:** Conceptualization, Methodology, Data curation, Software, Writing - Original draft. **Tero Kokkonen:** Conceptualization, Writing - Original draft, Funding acquisition.

Acknowledgments

This work was supported by the Regional Council of Central Finland/Council of Tampere Region and European Regional Development Fund as part of the Health Care Cyber Range (HCCR) project of JAMK University of Applied Sciences Institute of Information Technology. The authors would like to thank Ms. Tuula Kotikoski for proofreading the manuscript.

References

Alom, M.Z., Aspiras, T., Taha, T.M., Asari, V.K., Bowen, T., Billiter, D., Arkell, S., 2019. Advanced deep convolutional neural network approaches for digital pathology image analysis: A comprehensive evaluation with different use cases. arXiv preprint arXiv:1904.09075 .

Bera, K., Schalper, K.A., Rimm, D.L., Velcheti, V., Madabhushi, A., 2019. Artificial intelligence in digital pathology—new tools for diagnosis and precision oncology. Nature reviews Clinical oncology 16, 703–715.

van Diest, P.J., van der Wall, E., Baak, J.P.A., 2004. Prognostic value of proliferation in invasive breast cancer: a review. Journal of Clinical Pathology 57, 675–681. URL: https://jcp.bmj.com/content/57/7/675, doi:10.1136/jcp.2003.010777.

Dusenberry, M., Hu, F., 2018. Deep learning for breast cancer mitosis detection. https://github.com/CODAIT/deep-histopath/raw/master/docs/tupac16-paper/paper.pdf.

Feoktistov, V., 2006. Differential evolution. Springer.

Goodfellow, I., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples, in: International Conference on Learning Representations. URL: http://arxiv.org/abs/1412.6572.

Kang, X., Song, B., Du, X., Guizani, M., 2020. Adversarial attacks for image segmentation on multiple lightweight models. IEEE Access 8, 31359–31370. doi:10.1109/ACCESS.2020.2973069.

Khosravi, P., Kazemi, E., Imielinski, M., Elemento, O., Hajirasouliha, I., 2018. Deep convolutional neural networks enable discrimination of heterogeneous digital pathology images. EBioMedicine 27, 317–328.

Lin, B.C., Hsu, H.J., Huang, S.K., 2020. Testing convolutional neural network using adversarial attacks on potential critical pixels, in: 2020 IEEE 44th Annual

Computers, Software, and Applications Conference (COMPSAC), pp. 1743–1748. doi:10.1109/COMPSAC48688.2020.000-3.

Medical Image Analysis Group Eindhoven (IMAG/e), 2016. Tumor proliferation assessment challenge 2016. http://tupac.tue-image.nl/node/3.

Nasief, H., Zheng, C., Schott, D., Hall, W., Tsai, S., amd X. Allen Li, B.E., 2019. A machine learning based delta-radiomics process for earlyp rediction of treatment response of pancreatic cancer. npj Precision Oncology 3. doi:10.1038/s41698-019-0096-z.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A., 2017. Practical black-box attacks against machine learning, in: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 506–519. doi:10.1145/3052973.3053009.

Paul, R., Schabath, M., Gillies, R., Hall, L., Goldgof, D., 2020. Mitigating adversarial attacks on medical image understanding systems, in: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), pp. 1517–1521. doi:10.1109/ISBI45749.2020.9098740.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Price, K.V., 2013. Differential evolution, in: Handbook of Optimization. Springer, pp. 187–214.

Rajamäki, J., Nevmerzhitskaya, J., Virág, C., 2018. Cybersecurity education and training in hospitals: Proactive resilience educational framework (prosilience ef), in: 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 2042–2046. doi:10.1109/EDUCON.2018.8363488.

Schmitt, M.N., 2017. Tallinn manual 2.0 on the international law applicable to cyber operations. Cambridge University Press.

Sipola, T., Puuska, S., Kokkonen, T., 2020. Model fooling attacks against medical imaging: A short survey. Information & Security: An International Journal 46, 215–224. doi:10.11610/isij.4615.

Spanakis, E.G., Bonomi, S., Sfakianakis, S., Santucci, G., Lenti, S., Sorella, M., Tanasache, F.D., Palleschi, A., Ciccotelli, C., Sakkalis, V., Magalini, S., 2020. Cyber-attacks and threats for healthcare – a multi-layer thread analysis, in: 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), pp. 5705–5708. doi:10.1109/EMBC44109.2020.9176698.

Stokes, J.W., Wang, D., Marinescu, M., Marino, M., Bussone, B., 2018. Attack and defense of dynamic analysis-based, adversarial neural malware detection models, in: MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), pp. 1–8. doi:10.1109/MILCOM.2018.8599855.

Su, J., Vargas, D.V., Sakurai, K., 2019. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation 23, 828–841. doi:10.1109/TEVC.2019.2890858.

Su, J., Vargas, D.V., Sakurai, K., 2019. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation 23, 828–841.

U.S. National Cancer Institute at the National Institutes of Health (NIH), a. Cancer Incidence Rate. https://seer.cancer.gov/statistics/types/incidence.html.

U.S. National Cancer Institute at the National Institutes of Health (NIH), b. Cancer Statistics. https://www.cancer.gov/about-cancer/understanding/statistics.

U.S. National Cancer Institute at the National Institutes of Health (NIH), c. The Cancer Genome Atlas Program. https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga.

Veta, M., Heng, Y.J., Stathonikos, N., Bejnordi, B.E., Beca, F., Wollmann, T., Rohr, K., Shah, M.A., Wang, D., Rousson, M., Hedlund, M., Tellez, D., Ciompi, F., Zerhouni, E., Lanyi, D., Viana, M., Kovalev, V., Liauchuk, V., Phoulady, H.A., Qaiser, T., Graham, S., Rajpoot, N., Sjöblom, E., Molin, J., Paeng, K., Hwang, S., Park, S., Jia, Z., Chang, E.I.C., Xu, Y., Beck, A.H., van Diest, P.J., Pluim, J.P., 2019. Predicting breast tumor proliferation from whole-slide images: The TUPAC16 challenge. Medical Image Analysis 54, 111–121. doi:10.1016/j.media.2019.02.012.

Veta, M., Van Diest, P.J., Jiwa, M., Al-Janabi, S., Pluim, J.P., 2016. Mitosis counting in breast cancer: Object-level interobserver agreement and comparison to an automatic method. PloS one 11, e0161286.

Veta, M., Van Diest, P.J., Willems, S.M., Wang, H., Madabhushi, A., Cruz-Roa, A., Gonzalez, F., Larsen, A.B., Vestergaard, J.S., Dahl, A.B., et al., 2015. Assessment of algorithms for mitosis detection in breast cancer histopathology images. Medical Image Analysis 20, 237–248. doi:10.1016/j.media.2014.11.010.

Xu, H., Ma, Y., Liu, H.C., Deb, D., Liu, H., Tang, J.L., Jain, A.K., 2020. Adversarial attacks and defenses in images, graphs and text: A review. International Journal of Automation and Computing 17, 151–178. doi:10.1007/s11633-019-1211-x.

Zhang, W., Chien, J., Yong, J., Kuang, R., 2017. Network-based machine learning and graph theory algorithms for precision oncology. npj Precision Oncology 1. doi:10.1038/s41698-017-0029-7.