OULU 2015
C 528

ACTA UNIVERSITATIS OULUENSIS

Matti Mantere

NETWORK SECURITY MONITORING AND ANOMALY DETECTION IN INDUSTRIAL CONTROL SYSTEM NETWORKS

UNIVERSITY OF OULU GRADUATE SCHOOL;
UNIVERSITY OF OULU,
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING,
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING;
INFOTECH OULU

ACTA UNIVERSITATIS OULUENSIS
C Technica 528

MATTI MANTERE

NETWORK SECURITY MONITORING AND ANOMALY DETECTION IN INDUSTRIAL CONTROL SYSTEM NETWORKS

Academic dissertation to be presented with the assent of the Doctoral Training Committee of Technology and Natural Sciences of the University of Oulu for public defence in Auditorium TS101, Linnanmaa, on 29 May 2015, at 12 noon

UNIVERSITY OF OULU, OULU 2015

Acta Univ. Oul. C 528, 2015

Supervised by
Professor Juha Röning

Reviewed by
Professor Stephane Maag
Professor Ana Rosa Cavalli

Opponent
Professor Antonio Lioy

ISBN 978-952-62-0814-5 (Paperback)
ISBN 978-952-62-0815-2 (PDF)

ISSN 0355-3213 (Printed)
ISSN 1796-2226 (Online)

Cover Design
Raimo Ahonen

JUVENES PRINT
TAMPERE 2015

Mantere, Matti, Network security monitoring and anomaly detection in industrial control system networks.
University of Oulu Graduate School; University of Oulu, Faculty of Information Technology and Electrical Engineering, Department of Computer Science and Engineering; Infotech Oulu
Acta Univ. Oul. C 528, 2015
University of Oulu, P.O. Box 8000, FI-90014 University of Oulu, Finland

Abstract

Industrial control system (ICS) networks used to be isolated environments, typically separated by physical air gaps from the wider area networks. This situation has been changing and the change has brought with it new cybersecurity issues. The process has also exacerbated existing problems that were previously less exposed due to the systems' relative isolation. This process of increasing connectivity between devices, systems and persons can be seen as part of a paradigm shift called the Internet of Things (IoT). This change is progressing and the industry actors need to take it into account when working to improve the cybersecurity of ICS environments and thus their reliability. Ensuring that proper security processes and mechanisms are being implemented and enforced on the ICS network level is an important part of the general security posture of any given industrial actor.

Network security and the detection of intrusions and anomalies in the context of ICS networks are the main high-level research foci of this thesis. These issues are investigated through work on machine learning (ML) based anomaly detection (AD). Potentially suitable features, approaches and algorithms for implementing a network anomaly detection system for use in ICS environments are investigated.

After investigating the challenges, different approaches and methods, a proof-ofconcept (PoC) was implemented. The PoC implementation is built on top of the Bro network security monitoring framework (Bro) for testing the selected approach and tools. In the PoC, a Self-Organizing Map (SOM) algorithm is implemented using Bro scripting language to demonstrate the feasibility of using Bro as a base system. The implemented approach also represents a minimal case of event-driven machine learning anomaly detection (EMLAD) concept conceived during the research.

The contributions of this thesis are as follows: a set of potential features for use in machine learning anomaly detection, proof of the feasibility of the machine learning approach in ICS network setting, a concept for event-driven machine learning anomaly detection, a design and initial implementation of user configurable and extendable machine learning anomaly detection framework for ICS networks.

Keywords: anomaly detection, cybersecurity, industrial control system security, information security, intrusion detection, machine learning, network security

Mantere, Matti, Teollisuusautomaationverkkojen tietoturvan monitorointi ja anomalioiden havainnointi.
Oulun yliopiston tutkijakoulu; Oulun yliopisto, Tieto- ja sähkötekniikan tiedekunta, Tietotekniikan osasto; Infotech Oulu
Acta Univ. Oul. C 528, 2015
Oulun yliopisto, PL 8000, 90014 Oulun yliopisto

Tiivistelmä

Kehittyneet yhteiskunnat käyttävät teollisuuslaitoksissaan ja infrastruktuuriensa operoinnissa monimuotoisia automaatiojärjestelmiä. Näiden automaatiojärjestelmien tieto- ja kyberturvallisuuden tila on hyvin vaihtelevaa. Laitokset ja niiden hyödyntämät järjestelmät voivat edustaa usean eri aikakauden tekniikkaa ja sisältää useiden eri aikakauden heikkouksia ja haavoittuvaisuuksia.

Järjestelmät olivat aiemmin suhteellisen eristyksissä muista tietoverkoista kuin omista kommunikaatioväylistään. Tämä automaatiojärjestelmien eristyneisyyden heikkeneminen on luonut uuden joukon uhkia paljastamalla niiden kommunikaatiorajapintoja ympäröivälle maailmalle. Nämä verkkoympäristöt ovat kuitenkin edelleen verrattaen eristyneitä ja tätä ominaisuutta voidaan hyödyntää niiden valvonnassa. Tässä työssä esitetään tutkimustuloksia näiden verkkojen turvallisuuden valvomisesta erityisesti poikkeamien havainnoinnilla käyttäen hyväksi koneoppimismenetelmiä. Alkuvaiheen haasteiden ja erityispiirteiden tutkimuksen jälkeen työssä käytetään itsejärjestyvien karttojen (Self-Organizing Map, SOM) algoritmia esimerkkiratkaisun toteutuksessa uuden konseptin havainnollistamiseksi. Tämä uusi konsepti on tapahtumapohjainen koneoppiva poikkeamien havainnointi (Event-Driven Machine Learning
Anomaly Detection, EMLAD).

Työn kontribuutiot ovat seuraavat, kaikki teollisuusautomaatioverkkojen kontekstissa: ehdotus yhdeksi anomalioiden havainnoinnissa käytettävien ominaisuuksien ryhmäksi, koneoppivan poikkeamien havainnoinnin käyttökelpoisuuden toteaminen, laajennettava ja joustava esimerkkitoteutus uudesta EMLAD-konseptista toteutettuna Bro NSM työkalun ohjelmointikielellä.

Asiasanat: automaatiojärjestelmien turvallisuus, koneoppiminen, kyberturvallisuus, poikkeamien havainnointi, tietoturva, tunkeutumisen havainnointi

Preface

This work presented in this thesis was carried out in the service of VTT Technical Research Centre of Finland. The projects were funded by the VTT Technical Research Centre of Finland, TEKES (the Finnish Funding Agency for Innovation) and the European Commission. Projects included MOVERTI (Monitoring for Network Security Status), DIAMONDS (Development and Industrial Application of Multi-Domain Security Testing Technologies), ICS-Infosec (ICS Information Security), INCYSE (Industrial Cyber Security Endeavour) and SGEM (Smart Grids and Energy Markets). Important insights were also gained during the preparation phase and the first months of the ECOSSIAN (European Control System Security Incident Analysis Network) FP7 project which I led for VTT's part before moving to a new employer, Intel Security, during August 2014.

First of all, I wish to thank the supervisor of my thesis, Prof. Juha Röning for his guidance and assistance in completing this work. I also wish to thank the co-authors of the publications presented in this thesis: Mr. Mirko Sailio, Mr. Sami Noponen, Ms. Pia Olli, Mr. Jarno Salonen, and lastly Mr. Ilkka Uusitalo, who tragically and unexpectedly passed away during the Spring 2012. I also wish to thank my other close colleagues from the Cybersecurity Team in which I was a leader for the better part of the years 2011-2013: Mr. Pasi Ahonen, Mr. Heimo Pentikäinen, Dr. Kimmo Halunen, Mrs. Kaarina Karppinen and Mrs. Anni Karinsalo, Kimmo especially for his help during the writing process of this thesis and Pasi for his excellent work as my manager during the first years of my employment at VTT.

My gratitude also goes to the project managers, participants and the funding organizations of the aforementioned projects, not forgetting the overall support offered by the VTT and my manager, Mr. Hannu Honka.

I also wish to thank my parents, Mikko and Tiina, and my brothers Jukka and Tuomo with their families for providing the solid foundation and support that have helped me along through the years.

Finally, I wish to thank my wife, Elisa, my son Kaarlo and my yet-to-be-named daughter. Elisa for her patience, love and support during the occasionally trying times of the writing process. Kaarlo especially for his keen talent of occasionally providing welcome and enforced diversion, and the new baby for just being wonderful!

Abbreviations

f_n	*False Negative*
f_p	*False Positive*
d_{euc}	*Euclidean Distance*
d_{che}	*Chebyshev Distance*
AD	*Anomaly Detection*
AI	*Artificial Intelligence*
ANN	*Artificial Neural Network*
APT	*Advanced Persistent Threat*
ASCII	*American Standard Code for Information Interchange*
A-NIDS	*Anomaly based Network Intrusion Detection System*
CI	*Critical Infrastructure*
CLI	*Command-Line Interface*
COTS	*Commercial-of-the-Shelf*
DCS	*Distributed Control System*
DFA	*Deterministic Finite Automata*
DNP3	*Distributed Network Protocol*
EMLAD	*Event-driven Machine Learning Anomaly Detection*
GA	*Genetic Algorithm*
GP	*Genetic Programming*
GNP	*Genetic Network Programming*
HIDS	*Host-based Intrusion Detection System*
HMI	*Human-Machine Interface*
HMI-PLC	*Human-Machine Interface to Programmable Logic Controller*
ICS	*Industrial Control System*
IDS	*Intrusion Detection System*
IPS	*Intrusion Prevention System*
IP	*Internet Protocol*
IPv4	*Internet Protocol version 4*
IPv6	*Internet Protocol version 6*
IA	*Industrial Automation*
IACS	*Industrial Automation and Control System*

ICMP	*Internet Control Message Protocol*
IoT	*Internet of Things*
LAN	*Local Area Network*
MAC	*Media Access Control*
MBM	*Machine learning Bro Module*
M2M	*Machine-to-Machine*
NIDS	*Network Intrusion Detection System*
NSM	*Network Security Monitoring*
PERA	*Purdue Enterprise Reference Architecture*
PoC	*Proof-of-concept*
PCS	*Process Control System*
PLC	*Programmable Logic Controller*
RBM	*Restricted Boltzmann Machine*
SG	*Smart Grid*
SOM	*Self-Organizing Map*
SCADA	*Supervisory Control and Data Acquisition*
SVM	*Support Vector Machine*
TCP	*Transmission Control Protocol*
UDP	*User Datagram Protocol*
WLAN	*Wireless Local Area Network*
WAN	*Wide Area Network*

List of original publications

This thesis is based on the following articles, which are referred to in the text by their Roman numerals (I–VII):

I Mantere M & Uusitalo I & Sailio M & Noponen S (2012) Challenges of Machine Learning Based Monitoring for Industrial Control System Networks. Proceedings of 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2012).

II Mantere M & Sailio M & Noponen S (2012) Feature Selection for Machine Learning Based Anomaly Detection in Industrial Control System Networks. Proceedings of 2012 IEEE International Conference on Green Computing and Communications (GreenCom 2012): 771-774. Presented in 2nd workshop on Security of Systems and Software resiLiency (3SL 2012) organized during GreenCom.

III Mantere M & Sailio M & Noponen S & (2013) Network Traffic Features for Anomaly Detection in Specific Industrial Control System Network. (Future Internet 5(4)): 460-473.

IV Mantere M & Sailio M & Noponen S (2014) A Module for Anomaly Detection in ICS Networks. Proceedings of 3rd ACM International Conference on High Confidence Networked Systems (HiCoNS 2014).

V Mantere M & Noponen S & Olli P & Salonen J (2014) Network Security Monitoring in Small-scale Smart-grid laboratory. Published in the 2nd International Workshop on Emerging Cyberthreats and Countermeasures (ECTCM 2014).

VI Sailio M & Mantere M & Noponen S (2014) Network Security Analysis Using Behavior History Graph. Published in the Industrial Track Workshop at the 9th International Conference on Availability, Reliability and Security (ARES-IND 2014).

VII Mantere M & Sailio M & Noponen S (2014) Detecting Anomalies in Printed Intelligence Factory Network. Published in the 9th International Conference on Risks and Security of Internet and Systems (CRiSIS 2014). In press.

Contents

1 Introduction

The security of communication networks is an important part of the overall security posture of any given networked system. As of the writing of this text, more and more systems are getting connected in step with the progress of the process called the Internet of Things. The IoT paradigm holds that an increasing number of the physical devices will include computing capacity, with these devices being increasingly connected to the Internet. (Mattern & Floerkemeier 2010, Kortuem *et al.* 2010)

The increase in the connectivity of devices represented by the IoT continues unabated. This on-going change creates new security challenges, increases the size of the attack surfaces of systems, and thus highlights the need of addressing the issue of security. Systems and devices that historically used to be isolated from the wider area networks are especially vulnerable (Igure *et al.* 2006, Knapp 2011, Bruner 2013).

Several critical applications with implications ranging from the privacy of single individuals to the security of entire sovereign states rely on secure communications and their availability. There are several approaches to increase the security of communication networks to provide the required confidentiality, integrity and availability of transmitted information and services. One such approach is the monitoring of the network traffic and network state for malignant activity (Bejtlich 2004).

In a world where the data networks are nearly ubiquitous and used to provide critical and sensitive services, their security and general reliability are critical. The reliability of the data networks depends heavily on the proper security posture. There are no signs of external or internal attacks on network infrastructure and their components easing up in the near future. The claim of importance of data networks holds true both for the functions of the economy, as well as for the protection of different actors against crime and other malign influences projected through the various networks. (Bejtlich 2013, Anderson 2008)

Industrial control systems are a particularly good example of historically isolated systems that are now operating in an environment that has been rapidly opening up. In addition to other, less important operations, they are used to provide critical services, utilities and goods. A noteworthy security aspect of the ICS, and by extension their networks, is that a failure has the potential to cause serious physical consequences. The consequences depend on the severity of the failure and the processes that the ICS is

driving. It is therefore important to muster the needed network level defenses, such as network security monitoring (NSM) in addition to maintaining adequate security posture. (Nicholson *et al.* 2012, Knapp 2011)

Network security monitoring is a very broad subject, it has its place as an important part of the defensive measures of any organization or individual responsible for operating a data network (Bejtlich 2004, 2013, Knapp 2011). It can be divided into several topics with significant overlap between them and resulting in varying and occasionally even confusing definitions and terminology. The definition used in this thesis for general network security monitoring is defined by Bejtlich (2004) as *"The collections, analysis, and escalation of indications and warnings to detect and respond to intrusions"*.

Network security monitoring is not restricted to the monitoring of just the network traffic through taps or monitoring ports. However, we will restrict the subject scope in this thesis to encompass only the *network security monitoring based on the information extracted by capturing the network traffic*.

NSM provides part of the information needed for the creation and upkeep of situational awareness. The relatively lengthy delays in addressing known security critical bugs in ICS systems puts additional pressure on the non-proactive, or reactive, security systems. In paper by Pollet (July 2010), the researchers report that the average time it took for the vendors to fix a newly discovered bug in their systems was 331 days. Having a known and potentially exploitable bug in the system for that amount of time is a clear security concern. This existence of known vulnerabilities and added connectivity is a significant issue, and increased external attacks have been reported. (Okhravi & Nicol 2009)

The main environment and context for the anomaly detection work presented in this thesis is the ICS communication and control network, or simply ICS network. For the different networks, a three tiered definition is used for describing the level of isolation, or openness. The different network exposure level definitions used are as follows: isolated, restricted and open networks. The differences as defined in this thesis are explained in Table 1. These definitions represent simplifications. For example: true logical and physical isolation is not achievable, all networks do have physical and logical context of some form. One of the key issues in ICS and ICS network cybersecurity is the rapid change of system and network connectivity, specifically evolution from relatively isolated environments towards increased connectivity. (Alcaraz *et al.* 2013, Igure *et al.* 2006, Bruner 2013, Knapp 2011)

A given ICS network should be an example of a restricted, if not isolated, network

Table 1. Isolated, Restricted and Open Networks.

Type	Definition	Example
Isolated network	A network environment logically and physically separated from WAN with strictly controlled device deployment.	A traditional ICS network implemented without wireless connectivity and physically separated from other networks by an air gap. These networks are becoming exceptionally rare.
Restricted network	A network environment with strictly controlled and limited devices and connectivity to WAN.	A modern ICS network connected to the office network through demilitarized zone (DMZ) which in turn is connected to the Internet. Modern ICS networks also typically include various vendor maintenance access connections.
Open network	A network with little or no control over the connected devices or the connectivity to other networks.	Typical office or home user networks. The Internet is a good example of an exceptionally open network.

type. ICS networks used to be isolated, separated from the wide area network (WAN) by a physical air gap, but this is rarely the case anymore. Therefore, ICS networks typically represent the restricted category of networks, having lost their isolation. This makes them, at least partially, accessible to the outside world through various interfaces e.g., direct remote maintenance access through WAN, Corporate LAN or other routes. This also makes them more susceptible to cyber-attacks compared to their earlier isolated incarnations. (Nicholson *et al.* 2012, Knapp 2011, Bruner 2013)

Restricted ICS networks that have been operating for a long period of time typically used to be isolated networks. Most of the older systems were designed for operation in isolated environments, and the number of truly isolated networks is diminishing as air gaps are being phased out (Bruner 2013). This is at least partially due to the usefulness of the ICS data, and requirements for remote access possibilities such as cloud based SCADA components (Alcaraz *et al.* 2013, Bruner 2013). The security precautions of isolated networks were different from those of other networks, with a decreased level of network security monitoring and the total lack of peripheral network security monitoring, as there were no network edges connected to the outside world. The attack vectors into these isolated networks were primarily derived from the insider

threat (Knapp 2011). The definition of restricted network as presented in Table 1 is not very clear cut, the type encompasses a large variety of networks, as networks are very rarely truly open without any restrictions, or truly isolated without any paths to outside networks.

As the isolated networks are moving towards more openness and becoming restricted networks, it is important to consider new aspects of the network security resulting from the increased connectivity (Igure *et al.* 2006, Okhravi & Nicol 2009, Knapp 2011). The environments in which restricted networks are found typically demand high security precautions, a good example being an industrial site. Even with the shift from isolation towards more openness, the nature of the restricted networks enables specialized approaches to the security, or network security monitoring. Without deployment of proper NSM procedures, any cybersecurity situational awareness is undermined. Since the current trends point to processing power and network connectivity being integrated into more and more devices, monitoring these ubiquitous networks for security becomes more important, also in the case of ICS networks. However, even while the larger trend is towards less isolation, such as the advancement of IoT and industrial Internet (Mattern & Floerkemeier 2010, Bruner 2013, Beran *et al.* 2010), the ICS networks remain restricted, even if not isolated. (Alcaraz *et al.* 2013, Langner 2011a, Knapp 2011)

Several solutions for network security monitoring exist for different operational setting, including the various intrusion detection systems (IDS) widely deployed. Operating a network security monitoring system in a generic corporate or home network environment can be as easy as acquiring a commercial-of-the-shelf (COTS) solutions and deploying it according to the instructions provided. (Bejtlich 2004)

The situation is different for the restricted or even altogether isolated network environments. These environments, or at least, the more specialized sub-types such as ICS networks, typically exhibit special characteristics that negatively affect the feasibility of using standard COTS solutions without customization.

Even while it is the case that the ICS networks host increasing numbers of COTS solutions, considerable expertise is needed to tune a COTS security monitoring component, be it anomaly or intrusion detection system, to these or other restricted networks with special natures. COTS components of the generic network infrastructure found in industrial sites are also causing security issues by adding to the attack surface, this is true for both the software and hardware components (Igure *et al.* 2006). The increasing number of commodity systems found within ICS environments is therefore a

double-edged sword, bringing the operator benefits in the way of, e.g. cost savings and usability, but also adding to the security concerns. The esoteric and closed nature of the previous generation of ICS systems provided them a measure of security through obscurity. The COTS components, including their vulnerabilities, found throughout other industry sectors such as ICT, are well known to a much larger number of individuals and organizations.

For the generic anomaly detection setting, Callegari *et al.* (2013) provide the following statement highlighting some of the challenges: "AD in operational networks is a challenging task. Indeed, these are highly heterogeneous, complex and constantly evolving systems, where human errors, equipment misfunctioning and deliberate attacks[80] are non-negligible and mutually interacting events." Usage of machine learning based AD has had serious difficulties in the past. Coping with an open network with diverse and rapidly changing environment and traffic patterns remains a challenge when using machine learning based systems (Sommer & Paxson 2010).

However, systems such as presented by Kayacik *et al.* (2007) and Ramadas *et al.* (2003) employ ML, here namely SOM algorithm (Kohonen *et al.* 2001), for anomaly detection in general network settings. Kayacik *et al.* (2007) apparently use Bro logs as input for their hierarchical SOM which is implemented using SOM-PAK(Kohonen *et al.* 1996). Compared to the proof-of-concept SOM implementation presented in this thesis called MBM, Kayacik *et al.* (2007) use a different set of features and a hierarchical SOM. Few of the features are the same as used in the initial introduction of the MBM in paper IV, namely the duration of a connection and the data sent by originator and receiver of a connection.

Using machine learning based anomaly detection solutions for ICS network environment has been investigated by other parties as well, e.g. in Linda *et al.* (2009), Linda *et al.* (2011) and Linda *et al.* (2012) where the authors investigate different solutions for this purpose. Machine learning approach for anomaly detection is further discussed in more detail in Section 3. The work presented by them and, e.g. by Hadeli *et al.* (2009) who investigate the leveraging of the deterministic properties of ICS networks converge with the research presented in this thesis and support the idea of leveraging machine learning for anomaly detection in ICS networks.

The approach of using machine learning methods for anomaly detection in ICS networks appears to be a logical one, given the more deterministic, cyclic and stable nature of these networks (Hadeli *et al.* 2009).

For the work of this thesis, information was received from several sources: industrial

site network in Finland, smart-grid laboratory located at VTT Oulu, and most importantly, from a printed intelligence pilot factory Printocent PrintoCent, (accessed 1/6/2013) similarly located at VTT Oulu. The PoC solution presented in this thesis, created as an extension module for the open source network security monitor, Bro (Paxson 1999), was accomplished using the systems own scripting language. The module requires no changes to the Bro system, as is normal for Bro scripts.

The PoC system is currently called MBM as an acronym for Machine learning Bro Module. The system is a single algorithm and single event implementation of event-driven machine learning anomaly detection concept. In our anomaly detection and network security monitoring approach, we do not rely only on modeling the network traffic through different events, but also include the network state in the event-context tied machine learning models as well, as interpreted by Bro.

1.1 Terminology

The terminology concerning industrial automation systems is prone to being interpreted differently by different parties (Weiss 2010). There is overlap and confusion (Knapp 2011). It is therefore prudent to elaborate and define some of the main terms as used in this thesis. The definitions provided by Weiss (2010) are followed when not otherwise noted. As an exception to this, the original definitions are used when citing source material. The definitions from Weiss (2010) are used to provide a single base of definitions from a relatively recent work touching the main areas of the subject matter of ICS security.

Industrial control system, or ICS, is used to cover all industrial control and automation components, such as supervisory control and data acquisition (SCADA), and industrial automation (IA) systems. It is used as an umbrella term. ICS is also used as a synonym with industrial automation and control systems (IACS). The combined definition by Weiss (2010) for IACS and ICS is the definition for ICS followed in this work. According to Knapp (2011): "SCADA, or Supervisory Control and Data Acquisition, is just one specific piece of an industrial network, separate from the control systems themselves, which should be referred to as Industrial Control Systems (ICS), Distributed Control Systems (DCS), or Process Control Systems (PCS)".

SCADA is defined by Weiss (2010) as "Type of loosely coupled distributed monitoring and control system commonly associated with electric power transmission and distribution systems, oil and gas pipelines, and water and sewage systems. Supervisory

control systems are also used within batch, continuous, and discrete manufacturing plants to centralize monitoring and control activities for these sites."

Weiss (2010) defines ICS in the Appendix D of the book as follows: "Industrial control systems (ICS)[1] are an integral part of the industrial infrastructure providing for the national good. These systems include distributed control systems (DCS), supervisory control and data acquisition systems (SCADA), programmable logic controllers (PLC), and devices such as remote telemetry units (RTU), smart meters, and intelligent field instruments including remotely programmable valves and intelligent electronic relays.".

Further, Weiss (2010) defines IACS as follows: "information systems such as advanced or multivariable control, online optimizers, dedicated equipment monitors, graphical interfaces, process historians, manufacturing execution systems, and plant information management systems associated internal, human, network, or machine interfaces used to provide control, safety, and manufacturing operations functionality to continuous, batch, discrete, and other processes."

Depending on the time period, geographical location or personal preferences of researchers, these terms are somewhat interchangeable, although this too might be argued. The terms themselves are constantly changing, and are prone to being interpreted differently.

Industrial internet is another term that is affiliated with the ICS. It is in itself a very loosely defined term for which a consensus of its meaning is still being formed. Beran *et al.* (2010) discuss the applicability of general internet technologies for integration with the industrial automation. This usage of internet technologies in and for ICS systems is the definition used for industrial internet when discussed in this thesis. When defined this way, it is a sub-set of ICS networks. Industrial internet can also mean other things, such connecting various industrial devices into the Internet (Bruner 2013).

As a bottom line, industrial control system or ICS is used here to denote automation environments and their components with ICS networks that describe all the network environments used by and for the industrial control traffic.

It is also worth noting that the word *deterministic* is used interchangeably to the term *predictable* and neither term is used to denote a completely deterministic environment. A network with fully deterministic nature would have future states that could be accurately predicted from the current state. This is not the case and no such claim is made. The terms are merely used to describe an environment in which reliable estimates on the future states can be made, as long as there are no externally imposed changes.

The definition used in this thesis for network security monitoring is from Bejtlich

(2004) and goes as follows: **"The collections, analysis, and escalation of indications and warnings to detect and respond to intrusions"**. This definition of network security monitoring is broad and encompasses several different methods of information collection and processing. Therefore, the scope is additionally restricted to encompass only the monitoring done by capturing the network traffic. This is not to claim that the monitoring is the sole important factor of functional network security monitoring. The other methods of information retrieval also play an important role in the security monitoring of any given network. This definition of NSM used in this thesis also falls into the category of "Network access and interaction monitoring" as defined in Langner (2011a).

1.2 Contributions of the thesis

This thesis is formed of seven peer reviewed research articles. All of the articles are in the field of cybersecurity. The focus is specifically on the security aspects of industrial control systems networks, and by extension, critical infrastructure networks and networks that exhibit characteristics that are adequately similar.

The author of this thesis is the primary investigator and main driving force of the research work documented in articles I-V and VII. This includes driving the use of machine learning for anomaly detection in ICS networks, the usage of Bro NSM as the framework for implementing a proof-of-concept, and the eventual concept of event-driven machine learning anomaly detection approach. In article VI, the role was more supporting in nature, with focus on the connection of ML approaches and ICS networks to the papers subject matter. These and other main contributions are itemized in Table 2.

Network security monitoring in restricted network environments has gained momentum and increased in importance during recent years. ICS networks are an example of these types of networks. It has become apparent that these environments are experiencing a transformation from isolated networks to restricted ones, losing a part of their security protection brought by their relative isolation. The loss of network isolation and the relatively good protection offered by that status is currently an on-going process. However, the continued restricted nature of these networks allows for efficient monitoring to be deployed. They are not isolated, but certain environments are likely to remain restricted for the near future. (Bruner 2013, Langner 2011a, Knapp 2011)

This thesis contributes to the increased automation and simplicity for the deployment

of network monitoring and specifically anomaly detection solutions. The main emphasis is on the machine learning approach as exemplified by the proof-of-concept ML module implementation for the Bro Network Security Monitoring system. The main target environment is the general ICS network as an instance of a restricted control and communication network.

The original articles included in this work were published between 2011 and 2014. The work presented in the papers was focused on restricted networks and industrial site cybersecurity.

For the security of restricted networks and in ICS networks in particular, an emphasis was placed on network security monitoring on IP level and the use of machine learning approaches, namely SOM, in the latter stages of the work. These issues are mainly discussed in Publications I, II, III, IV and VII. A PoC system implementation is presented in the latter two.

Publication I discusses the applicability of using machine learning based methods for network security monitoring in ICS environments. The paper presents the early stages of the work leading to this dissertation. A set of possible features that could be evaluated later on for possible use is also discussed. A three tier mapping of ICS environment is used in this paper. This publication was an initial publication describing the idea. The main contribution is a possible feature set together with analysis and indication of feasibility of the approach in ICS environments. Discussion on the matter had and has been ongoing by other researchers as well, providing a good starting point. Initially, an existing VTT machine learning system implementation was planned to be used, but this idea was later discarded in favor of using the Bro NSM system and new algorithm implementations.

Publication II discusses the applicability of the possible features presented in Publication I to a particular given industrial control system network. The ICS network in question is located at a large Finnish industrial site. The exact location and nature of the site cannot be divulged due to the confidentiality of the matter. A network capture from the site was analyzed and network structure used to draw conclusions. The main contribution was the rationale and selection of a feature set for use in ICS networks, and the network under investigation in particular.

Publication III further expands and elaborates the issues discussed in Publication II, Publication II being a conference paper and Publication III an expanded journal article. More data, discussion and information, as well as general refinement to the conclusions and deduction is presented in Publication III. In addition, the applicability of certain

features, possible new ones and general approach are further researched. A contribution of this paper includes an extended and more fleshed out feature set, including a strong continued indication of the feasibility of the approach. The author of the thesis and the paper was not aware of a set with the same features being discussed for the type of environment in question. This also holds true for paper II.

Publication IV describes an extension built on top of Bro Network Security Monitor System (Paxson 1999) implementing a SOM algorithm and several features. The system is aimed at noticing anomalies in real-time on restricted or ICS network which exhibit sufficient determinism and uses network protocols which can be parsed by the underlying Bro NSM system and have sufficient features implemented for that particular protocol. The main contribution of the work presented in this paper is the introduction of an easy to use and user configurable proof-of-concept system for anomaly detection in ICS settings. No such system implementing SOM with the specific feature set used, aimed for use in ICS networks, and using the Bro system for support was known to the author during the time of publication.

Publication V presents initial investigations into the network security monitoring issues inherent in smart-grid environments. The publication describes the initial network security monitoring setup for a smart-grid laboratory. This work presents early steps for leveraging the environment in investigating network security monitoring in other ICS environments than Printocent (PrintoCent, accessed 1/6/2013). In addition to the monitoring, privacy, vulnerabilities and security related standardization are discussed. The overall paper preparation, and especially the monitoring issues, were the responsibility of the first author.

Publication VI presents a novel method for network security analysis using a behavioral history graph. The author of this thesis is the second author of that paper. The role of the author in the paper consisted of assisting the primary investigator in formulating the approach, providing context knowledge for industrial control system networks and machine learning based anomaly detection approaches. The novel method presented is aimed for usage in ICS networks and has been investigated for use in combination with more fuzzy approaches, such as the MBM system presented in Publication IV.

Publication VII presents a preliminary validation of the system described in Publication IV as applied to a novel environment presented by the Printocent printed intelligence MAXI -line PrintoCent, (accessed 1/6/2013). The SOM module is first trained using the captured data traffic, and tested using simple anomaly cases. Embedded anomalies were

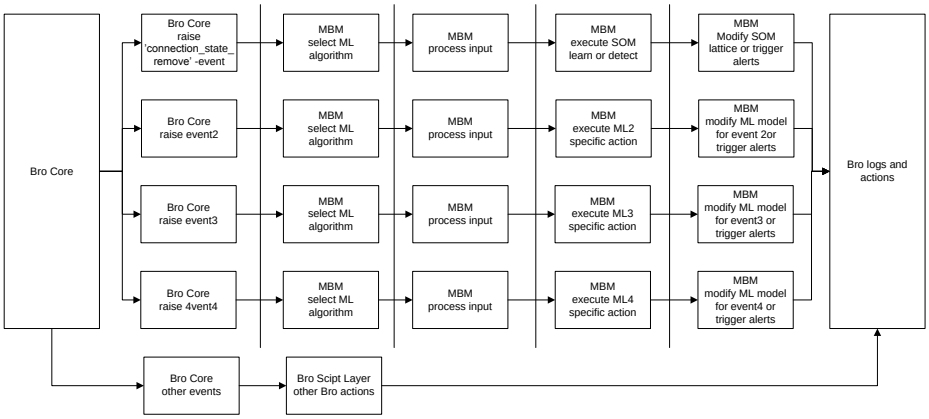

Fig 1. Event driven architecture.

reconnaissance activity traffic generated by Nmap (Nmap Network Security Scanner (accessed 7/2/2013), Nessus (Nessus Vulnerability Scanner (accessed 1/2/2014) and Nikto (Nikto2 Web Server Scanner (accessed 2/3/2014) scanning activity. Contribution of this work is the validation of the selected approach of machine learning anomaly detection for ICS network, and secondarily of the system and demonstration of the performance as documented in the publication. The event-driven machine learning approach is gradually introduced in IV and VII. The concept, a part of which is used in MBM architecture as a PoC, is elaborated in Figure 1.

The current state of the implementation of the ML module for Bro NSM, is presented in Chapter 4.

1.3 Research questions and approach

The high-level research question of this thesis is to investigate a way to improve the cybersecurity state of ICS and CI networks. More specifically: **Whether machine learning could be leveraged to improve the automation level and decrease the manual customization necessary for deployment of anomaly detection solutions in these networks.** The hypothesis is that one of the main issues holding proper NSM back in ICS environments is the difficulty in their customization and deployment. This situation could be ameliorated through increased automatic customization by using machine learning to model the existing network environment. Anomaly detection is seen as a component of network security monitoring.

The fact that even well-functioning anomaly detection is not sufficient in itself to secure a network is realized. However, the possible increment in NSM capability and overall network security with the possibility of zero-day detection were deemed sufficient impetus to target the anomaly detection part of NSM.

One of the core issues with security status of different ICS systems today is their increased connectivity and therefore significantly increased surface area of their attack planes (Ericsson 2010, Igure *et al.* 2006). The vulnerability and high damage potential for security breaches of connected industrial control systems and critical infrastructures create a potentially very dangerous situation. The exploits directed against ICS and critical infrastructures (CI) are available to a large number of individuals, organizations and nation states. No international regulations whatsoever are in place to control the availability of and access to these exploits and their delivery mechanisms, nor are there likely to be any in the short term. Whether such regulations and possible restrictions should or could be implemented is not clear, for that matter, either. However, the proliferation of these exploits with their delivery mechanism and similar systems as a means of war, sabotage, hacktivism and terrorism is driving a major push to improve the resilience and general robustness of these systems.

A good, if already rather worn down example of the high damage potential of deployed offensive cyber defense capability or an advanced persistent threat (APT) attack, would be the Iranian Stuxnet incident (Langner 2011b). The book by Langner (2011a) discusses the post-Stuxnet cybersecurity situation of industrial control systems. The Stuxnet was perhaps the first widely publicized example of a state or a group of states deploying advanced cyber weaponry against the security interests of another country (Nicholson *et al.* 2012). Knapp (2011) states that "There is no doubt about it at this time: Stuxnet is an advanced new weapon in the cyber war.". The incident briefly highlighted the weak security situation in the automation world, but the effect proved to be mostly temporary. The weak security posture prevalent in the ICS domain is likely to continue. (Langner 2011a)

Regardless of the context of the conflict and the motive of the adversaries, a correct situational awareness, especially in the context of cybersecurity, is important. This is true for planning phase of defensive posture and for executing operational defense. Network security monitoring as exemplified by intrusion detection and prevention, as well as anomaly detection form an important part of the overall resilience and situational awareness for networked equipment. This is also true for the various ICS and CI environments.

The need of network security monitoring and overall increased cybersecurity for ICS systems and CI in general has been apparent for a while (Nicholson *et al.* 2012, Dzung *et al.* 2005). The high-profile incidents concerning industrial installations being targeted for cyber-attacks serve to raise the awareness level of the relevant public and private actors. During our investigations, other advances have been made in the field. However, the situation remains sub-optimal concerning the security level of the various installations around the globe. Good examples of such installations are, e.g. electrical infrastructures, and in this context, smart-grids in special. (Yan *et al.* 2012, Ericsson 2010, Moslehi & Kumar 2010)

The hypothesis that network security monitoring and anomaly detection for ICS networks could leverage the determinism, cyclic nature and stability inherent in most of them has been gaining scientific backing in the literature. This provides the motivation for further investigation into the issue. The investigations on machine learning based NSM method to fully utilize these special attributes was chosen as a starting point to investigate the matter. It also follows that the same approach would likely be feasible for any network with sufficiently similar attributes, namely restricted or closed network environments with specialty hardware and software of suitable nature.

The issues inherent with in a machine learning approach for anomaly detection in communication networks are highlighted in (Sommer & Paxson 2010). Initial investigations pointed to a logical conclusion that given these constraints, it should be feasible to construct such a system for more restricted or ICS environments since most of the objections did not touch that particular subset of network environments. A similar conclusion has also been reached by other authors as well, e.g. in papers by Linda *et al.* (2009, 2011, 2012), Hadeli *et al.* (2009).

Interestingly, the Stuxnet cyber weapon investigated in Langner (2011b) could possibly have been detected using a machine learning based anomaly detection system. A sufficiently advanced system that would have kept track of the actual control values transmitted by the infected control system through the respective protocols to the actuators, or centrifuges in this case, could have noticed the destructive out-of-bounds parameters. However, this would have required advanced learning with the system looking at the ICS protocol specifics and the payload control parameters. Given correct and available protocol parsers and a system that would have had the time to learn the normally occurring parameter boundaries, the transmitted out-of-bounds values could have perhaps been possible to catch.

Initially, we focused on the challenges of using such a system in the ICS context

on a high-level of abstraction and then continued to investigate the approach on real ICS network data captures. During these investigations, it was chosen to develop and implement a machine learning extension for an open-source NSM solution using a suitable algorithm. For the initial approach, SOM was chosen as the algorithm and Bro NSM as the open-source platform. An important point is the decision that the system would need to work well enough to provide additional benefit for network security monitoring activities with a limited extra effort. Catching everything with this type of solution was seen as unrealistic at best. Eventually, the implemented system came to exist as a specific case of an event-driven concept currently documented briefly in paper IV and with some more detail in paper VII and with a dedicated publication in the works. The event-driven concept remains one of the major contributions of this thesis.

One of the important research questions faced by machine learning based anomaly detection systems is to identify the set of features used by the algorithm (Bengio *et al.* 2013). The optimal feature selection or the representation of the information for the machine learning algorithm used varies depending on the system to be modeled by it. For the network security purposes, this essentially boils down to a question of what is the smallest, most efficient set of features that can be used to clearly model the normal traffic while retaining good enough sensitivity to respond to the possible anomalies and minimizing (p_{fn}/p_{tn}) and (p_{fp}/p_{tp}). Where fn, tn, fp, tp represent false negative, true negative, false positive and true positive, respectively. The high rate of false positives and possible false negatives of generic learning IDS systems has gathered criticism such as presented by Rexworthy (2009). The task of minimizing both false positives and false negatives needs to be addressed but is difficult for a machine learning based system in an open and dynamic environment Sommer & Paxson (2010).

The tool that was widely used in this research, Bro NSM, became a tool of choice for the author while working at the International Computer Science Institute in Berkeley, California as a visiting short term scholar and working on the system internals. At that time, Bro NSM system had not yet been extended to function with ICS communication protocols, such as Modbus. However, that is changing with recent published work presenting new protocol parsers such as DNP3 and the release of Bro 2.2 implementing support for both DNP3 and Modbus protocols.

Extending the system with a machine learning module to model the deterministic traffic in the restricted environment seemed like a good option. Similar systems existed, targeted at more open networks and working on different modules. The availability of ICS protocol parsers enable the machine learning approach to utilize protocol specific

features, providing additional functionality and sensitivity. During the autumn 2013, at least two ICS protocol parsers became available for Bro NSM, namely Modbus/TCP and DNP3.

The literature review conducted during the investigations has revealed that several papers had been published on leveraging attributes such as determinism to increase the effectivity of NSM solutions in these environments.

We proceeded to publish some of our initial findings and ideas in conference papers by discussing the challenges of the approach and conducting case studies based on packet captures from industrial sites. Eventually, the research lead to the initialization of a development of PoC machine learning extension module for Bro NSM as planned. The initial machine learning algorithm selected was the Self-Organizing Map algorithm, also called Kohonen's map (Kohonen *et al.* 2001).

Table 2. Contributions.

Contribution	Notes
An indication of machine learning based anomaly detection approach feasibility in ICS networks.	Investigated and results presented in papers I, II and III. Using machine learning approach was judged a feasible option through research and literature review. A smart-grid dimension is investigated in paper V.
A set of possible features identified for anomaly detection in ICS networks.	Investigated in papers II, III and IV, based on analysis of real networks. Multiple features were investigated and findings reported.
A design for easily configurable NIDS based machine learning anomaly detection system.	A functional and easily configured PoC system for anomaly detection using SOM for restricted networks. Built as an extension for Bro NSM using the scripting language of the system. No such system implemented in Bro NSM script was known to the author before this. Initially presented in paper IV with the leveraged learning method for ML based AD presented in more detail in VI.
A demonstration that the system and approach work in real-life setting.	Tested using Printocent pilot factory network traffic in detecting embedded anomalies. Presented in paper VII. Attack-like behavior embedded in the captures for testing represented reconnaissance activity. The extra activity was detected.
A confirmation that ICS networks exist for which a machine learning approach is feasible.	Investigated using Printocent pilot factory network. Results of the investigation were summarized in papers IV and VII. The network exhibited some sub-optimal properties, e.g. varying operational time each day. The network traffic was still predictable enough to allow for machine learning approach.
A concept of event-driven machine learning anomaly detection.	The concept of a system with separate, optionally different, machine learning algorithm instances for each separate network event flow regardless of the network layer level of the event. The concept was initially sketched in paper IV and elaborated in slightly more detail in paper VII.

2 Network security monitoring and ICS networks

Network security monitoring can be used as an information collection part of active countermeasures aimed at disarming intruders who either have gained or are about to gain access to the monitored network. Catching the early signs about an intrusion attempt or an exploitation in its early phases may be enough to stop the attack, thus limiting damages or possibly to prevent the actual exploitation phase ever taking place. In addition to this, the logs and information collected can be used for forensic purposes after the active phase of the attack has subsided or been dealt with. The mitigation of the aftermath by discovering the flaws that lead to the compromise needs to be identified and dealt with, and there, the collected data and logs are invaluable. (Bejtlich 2004, Anderson 2008)

The network security monitoring in ICS networks includes monitoring such as the following communications: machine-to-machine (M2M) communication between different automation components, communications between other automation components and HMI devices, communication over network enclave edges and relevant remote connections. The communication infrastructure and therefore network security monitoring are required by automation environments for the continued production. The operational requirements and vulnerability of the network infrastructure typically go hand in hand, especially in larger or distributed ICS environments. (Langner 2011a, Knapp 2011, Weiss 2010)

In this work, we will mainly focus on the anomaly detection sub-topic of network security monitoring (Thottan & Ji 2003, Callegari *et al.* 2013). Even more particularly, the part which is relevant to the anomaly detection in ICS networks is targeted. In comparison to anomaly detection, network security monitoring encompasses a very large field of different methods and techniques aimed at providing maximal knowledge on the workings of the network from the security perspective. Most of the aspects of the network security monitoring are out of the scope of this thesis, but a good treatise on the subject can be found in (Bejtlich 2004, 2013).

Network security monitoring for ICS networks deals with a large variety of different automation related technologies. For example, in a smart-grid (SG) environment, there can be several significantly different sub-environments where entirely different

monitoring approaches might be warranted (Yan *et al.* 2012). In a typical factory site network, we have a segmented network structure with ICS zone, demilitarized zone and the corporate network zone. However, such segmentation could be difficult for an SG or similar environments that are significantly dispersed geographically and thus different NSM methods would be needed. A specially interesting issue for monitoring SG communication and control networks is the fact that the end-user can have physical control over the smart meter assigned to his residence and be able to manipulate it (Liu *et al.* 2009). The diversity between separate ICS environments leads to extensive tailoring and customization requirements if a given NSM system designed for a particular ICS network should be moved to another one.

The topics of intrusion detection and prevention, as well as anomaly detection are discussed in their respective sections, namely Sections 2.2 and 2.3. This is done for improved readability in the full knowledge that anomaly detection and intrusion detection significantly overlap. There is also variance in the literature, in paper by Tsai & Lin (2010) intrusion detection is used, whereas in paper by Shon & Moon (2007), effectively the same issue is discussed as anomaly detection.

A common denominator for various network security monitoring approaches is that they rely on gathering network information typically through network taps or the monitoring ports of network equipment when available. Some systems also work in-line and might include traffic blocking functionality.

Gathering the information through taps and monitoring ports both have their advantages and disadvantages. The actual tools and deployments that might be used for general NSM are not in the focus of this thesis, and are discussed fairly sparsely. Tools, e.g. Bro (Paxson 1999), Snort (Snort, accessed 1/12/2013), Tcpdump (Tcpdump, accessed 7/6/2013), ELSA (ELSA, accessed 1/4/2014), Wireshark (Wireshark, accessed 2/5/2013), Suricata (Suricata, accessed 1/12/2013) are all part of the family of tools that can be used to construct a proper NSM platform. The number of tools available is fairly large, and we limit our scope in this thesis to the tools with immediate relevance to our specific approach.

Noteworthy issues concerning network monitoring and ICS networks are the important differences in privacy concerns in relation to, e.g. office networks. The legislation controlling the monitoring activities of organizations on their employees differs greatly between countries. However, as the ICS networks are or should be void of personal communique such as emails that are typically sent from office workstations, the monitoring deployments are not restricted by these factors to the same degree. The

privacy concerns in an ICS networks have more to do with the sensitive information such as production control values and blueprints belonging to the entity typically operating the industrial site.

These privacy concerns need also to be adequately noted, but they are of a different nature. The monitoring activities need to be designed in such a manner that no unauthorized third party gains access to the logs or has the ability to view the traffic. Therefore, the privacy concerns are not so much diminished as they are changed in nature. The privacy of the individual becomes the privacy of the legal entity owning the information flowing in the network.

The network privacy aspects are also heavily affected by the local legislature, and monitoring the email traffic of individuals could be illegal even if using email from that network location is explicitly forbidden. The legal aspects of the privacy concerns are left out of scope in this thesis.

2.1 Network security monitoring infrastructure

Depending on the network to be monitored, specialized hardware might be needed for successful implementation and deployment of NSM mechanisms. Careful attention needs to be given to what is needed to be captured from the network, from what location and with what tools. Collecting everything from every location would be an idealized scenario, but typically not a feasible one. (Bejtlich 2004)

Monitoring large network environments requires either new hardware to be deployed or the exploitation of existing infrastructure components when feasible. Networks with high throughputs further require adequate computing power to be able to process everything in real-time. Distributed networks create their own additional requirements.

For a small-scale environment such as the Printocent network used as a test bed, a limited amount of hardware was required: an up-to-date laptop computer with sufficient computing power and three Ethernet ports, network tap and some cabling was all that was needed. Should more than one location need to be monitored, additional taps and a system with additional Ethernet ports would have been needed.

The level of equipment necessary for successful NSM deployment correlates with the needed level of observation and the size and architecture of the monitored network. If all the traffic is to be captured with all the noise, an inline tap for every cable would be needed, as explained in the Section 2.1.2. If a more rough capture is sufficient, a single monitoring port interface or a limited number of inline taps would possibly do, the exact

number of monitoring ports or taps depending on the architecture of the network and the desired level of traffic visibility. (Bejtlich 2004, 2013)

Network security monitoring typically requires the usage of various different systems or tools. A good example of an NSM system leveraging several existing tools is the Security Onion Linux distribution (Security Onion, accessed 14/5/2014). The system consists of network sensors such as Bro and Snort, log handling through a piece of software called Enterprise Log Search and Archive, or ELSA (ELSA, accessed 1/4/2014), and various other tools for NSM. The distribution assists in the configuration of the system for both standalone and distributed modes.

2.1.1 Monitoring ports

If monitoring ports are available in the switches used in a particular network, it is a straightforward task to connect the monitoring systems into the available ports and start monitoring the traffic. The monitoring port provides copies of the packets it sees passing through, regardless of for whom they are headed for. This allows for a centralized monitoring at one location. Different naming conventions for the monitoring ports exist, one example being the Switched Port ANalyzer (SPAN) port as referred to by Cisco Systems.

The monitoring ports do suffer from shortcomings. They are prone to dropping parts of the traffic, and filtering some of the noise out and thus not providing duplicates of all traffic for the monitoring port. For monitoring purposes, this is not an optimum solution, in most cases, it would be better to be able to see all the traffic. (Zhang & Moore 2007, Bejtlich 2004)

However, using monitoring ports is a straightforward task, it does not cause operational disturbances and requires no additional network hardware deployment if the feature is found in the installed systems.

2.1.2 Inline taps

Using inline taps to capture network traffic is typically preferable to the monitoring ports, if deployment of the taps is possible. Due to their inline nature, the taps require that the connection is routed through the tap, causing operational issues for the duration of the installation. The taps are typically transparent to the network, and do not cause significant, or even measurable latencies, when operational. Network taps are available

which are fail-safe in a way that a power loss or most equipment failures will only render the monitoring capability off-line, but the traffic will continue to pass through unhindered. This is not an attribute found in every network tap, and the actual tap to be deployed should be carefully selected. (Bejtlich 2004)

2.2 Intrusion detection and prevention

Intrusion detection is defined in the paper by Mukherjee *et al.* (1994) as follows: "The goal of intrusion detection is to identify, preferably in real time, unauthorized use, misuse, and abuse of computer systems by both system insiders and external penetrators."

This definition is very broad, practically encompassing the whole field of reactive information security. However, this is the overreaching goal that intrusion detection works towards.

Similar definitions as in the paper by Mukherjee *et al.* (1994) can also be found elsewhere with slightly varying wording and weights. Intrusion prevention systems (IPS) are here defined as systems with the capability to take active measures against the intruders' progress based on the information provided by sensors. Most IDS's are also in fact IPS's due to their structure and functionality, which provides active measures for manipulating the network environment or hosts therein. As a term, an IPS might be easier to market, but the difference between an IPS and IDS is typically just that. This is not an point of view shared by everyone, as presented by Rexworthy (2009) in his criticism aimed at IDS systems.

In this section, we focus mainly on the network-based intrusion detection, even while noting the potential for increasing efficiency by leveraging host-based context as presented in the paper by Dreger *et al.* (2005).

The early work on intrusion detection model that is presented in Denning (1987) formed a very important starting point for general intrusion detection research. The host based detection gradually came to be accompanied by network based intrusion detection. The early stages of the research are discussed in the paper by Mukherjee *et al.* (1994).

Network intrusion detection systems (NIDS) attempt to detect attacks and intrusion by looking at the network traffic. There are several approaches which can be used. Good examples of different approaches are provided by Bro and Snort. Bro NSM (Paxson 1999) represents an event based intrusion detection approach when used as a NIDS and Snort (Snort, accessed 1/12/2013) a signature based system. Other approaches exist as

well.

The signature based approach is much like a conventional virus detection system, a database of attack or intrusion patterns is continually compared against the perceived network traffic. This signature based approach is essentially black listing known misuse events as they are perceived in the network traffic.

In the event based system, the system is instructed to act upon certain events, such as triggering logging activity whenever a connection is terminated or using a regular expression when a chunk of TCP data is reassembled to discover interesting information.

Several commercial intrusion detection and prevention systems are currently available for purchase from several vendors. Some of these systems are proprietary with, while others, such as Bro (Bro NSM, accessed 2/12/2013), Snort (Snort, accessed 1/12/2013) and Suricata (Suricata, accessed 1/12/2013) are open source with varying degrees of commercial support available. Rules and usage of Snort for monitoring ICS networks, especially the Modbus/TCP protocol, are introduced in the work presented in the papers by Morris *et al.* (2013, 2012). As the protocol parsers for these also exist for Bro, using the information presented in these papers should be rather straightforward.

2.3 Anomaly detection

Anomaly detection in general is used in a wide variety of applications. Chandola *et al.* (2009) provide a comprehensive survey of different general anomaly detection methodologies. This also includes the various machine learning approaches. The work is not specific to network anomaly detection. However, it is a good starting point for discussing the applicability of anomaly detection for general network security monitoring purposes and further into the ICS network domain. In general, network anomaly detection is meant when discussing anomaly detection in this thesis, if not otherwise noted. Chandola *et al.* (2009) define anomaly detection as follows: "Anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior."

For network anomaly detection, we need a more contextual definition. Anomaly detection is not synonymous to network security monitoring, even if it is a part of it. It is also noteworthy that the definition of an anomaly can vary between individuals and different environments, even in communication network context.

In this thesis, we define anomaly detection as **detecting events or states of the network that differ from those historically seen**. This definition closely follows that

provided by Anderson (2008), Bejtlich (2004), Callegari *et al.* (2013) and Knapp (2011). Anomaly detection is not a clear-cut separate case from intrusion detection and prevention, but overlaps with it considerably. Our approach to anomaly detection is mostly placed in one of these overlaps with intrusion detection, as that is its primary intended functionality in the work presented; to detect intrusions and on-going attacks. We limit the scope on anomaly detection to the network-based anomaly detection, leaving out host-based approach or generic anomaly detection.

Detection of equipment failure or impending failure in the network environment is also an interesting possibility for detection with the right algorithm and the right set of tools. In this work, this area is defined as out of the scope.

In the work presented by Thottan & Ji (2003), network anomaly detection is divided into four most common categories at around year 2003: rule-based, finite state-machine-based, pattern matching-based, and statistical analysis-based approaches. The machine learning based approach overlaps significantly with the statistical analysis and pattern matching based approaches and shares some attributes with them.

Rule-based approach systems are typically very resource consuming, and not well suited for real-time operation due to their resulting slowness. Also, the rules need to contain knowledge about the network environment and its faults. The requirement of prior knowledge about the possible faults of the network is a difficult issue in itself, regardless of the potential performance. (Thottan & Ji 2003)

The finite state-machines approach is defined as: "Anomaly or fault detection using finite state machines model alarm sequences that occur during and prior to fault events" Thottan & Ji (2003). This form of an approach is used in the work presented in the paper by Goldenberg & Wool (2013) for Modbus/TCP protocol. In this approach, the possible protocol, network or system states are modeled. Deviations from the accepted states, incorrect sequence of states, impossible combination of states or a state not existing in the model typically are flagged as representing anomalies.

A statistical approach such as principal component analysis (PCA) can been used for anomaly detection as well, as is demonstrated by Issariyapat & Fukuda (2009). PCA could also be used as a part of a machine learning-based approach, but when used by itself, it belongs to the statistical approach category. Work presented in Honda *et al.* (2008) and Wagner & Plattner (2005) also investigates a statistical approach for anomaly detection, based on entropy of traffic, such as the packet headers.

In the Thottan & Ji (2003), the fourth group is named as pattern matching approach. In the pattern matching approach, the aim is to discover deviations from the historically

perceived network traffic patterns. The machine learning approach for anomaly detection can also make use of this pattern based approach and it can be difficult to reasonably separate the two in all instances. The machine learning-based approach also shares some attributes with the statistical approach to the extent that considerable overlap may occur as well as shared terminology. General machine learning in itself can also be viewed from a statistical point of view, as evident in the book by Murphy (2012).

An interesting approach by modeling single human-machine interface - programmable logic controller (HMI-PLC) Modbus/TCP channels aimed at intrusion detection work presented in (Goldenberg & Wool 2013). They demonstrate the use of a learning system, a finite state machine, that leverages the deterministic and finite nature of single HMI-PLC channels. As the single HMI-PLC channels are very deterministic, this approach is promising. Also, the rationale in using a learning system and leveraging communication determinism converges with our own hypothesis that using machine learning in ICS environments is feasible. This approach represents a reverse direction compared to our top-down approach. Where our system presented in Chapter 4 deals with the issue through network-wide monitoring, the work presented in Goldenberg & Wool (2013) starts by looking at individual communication channel of an individual protocol. However, work for extending the MBM to also handle modeling of much more fine-grained traffic such as single HMI-PLC channel using a suitable algorithm is on-going.

Learning algorithms of some sort do play a role in a number of different generic and network specific anomaly detection Chandola *et al.* (2009). Anomaly detection in connection with machine learning is discussed in more detail in Section 3.

2.4 ICS networks

ICS networks are the infrastructure which the automation and control equipment use to communicate. Plain text communication is historically common, as evident by the traffic analyzed in papers III and VII. The integrity and availability of this communication infrastructure is critical to the sustained operation of the ICS installation. The ICS equipment tend to lack the robustness typically expected from networked devices in more open environments with little capability to survive changes in their operating environment, such as their communication network, without adverse effects (Langner 2011a). They have traditionally been protected through physical means, in contrast to the digital measures taken in the ICT sector (Knapp 2011).

The hypothesis that network traffic is sufficiently predictable in ICS networks to provide a setting for using machine learning approach has been discussed in work presented, e.g. by Hadeli *et al.* (2009), Linda *et al.* (2009), Yang *et al.* (2006). Bruner (2013) also mentions the feasibility of deep packet inspection and anomaly detection given advances in big data analytics. Similar reasons for the feasibility of a more general anomaly detection approach are given by Knapp (2011).

Authors of the paper by Goldenberg & Wool (2013) demonstrate that a single HMI-PLC channel of Modbus/TCP communications can be modeled as a deterministic finite automata (DFA). They also note that SCADA systems have established and clear communication patterns. Authors of the work presented in (Valdes & Cheung 2009) also note this, by stating that "We argue that the model-based approach is more feasible for process control systems, because these systems tend to have a small and static set of applications, regular and predictable communication patterns, and simpler protocols". A number of key attributes for leveraging machine learning for anomaly detection in ICS networks are identified and are briefly explained in Sections 2.4.2, 2.4.1 and 2.4.3.

Figure 2 depicts a simplified example of a segmented network architecture for an industrial site with three network segments, the segments being ICS network, demilitarized zone and the office, or corporate, network. It should be noted that the example is simplified and not mapped to the Purdue Enterprise Reference Architecture (PERA) (Williams 1994) hierarchical levels. The simplified model is sufficient for the needs of this work and the PERA is not discussed in more detail. However, that is not to be taken as a statement against the relevance or importance of the PERA.

The ICS network contains the various equipment necessary for control of the processes under supervision. These include e.g. SCADA systems, human-machine interfaces (HMI's) and field devices such as RTU's, PLC's, instrumentation and actuators.

The demilitarized zone typically containts systems such as history servers, various file servers, ERP interfaces and the like. Production DMZ architecture is primarily suited for high security requirement environments Dzung *et al.* (2005). All the connections from the ICS network or the office network are terminated in the DMZ, no direct connections between the ICS network segment and the office network segment are to be allowed. The different functions inside the ICS network should further be compartmentalized into secure perimeters or enclaves with a necessary level of isolation from the other parts of the ICS network. The nature of the various perimeters and the functions they host should ideally dictate the types of security measures deployed at their borders or within. (Knapp 2011)

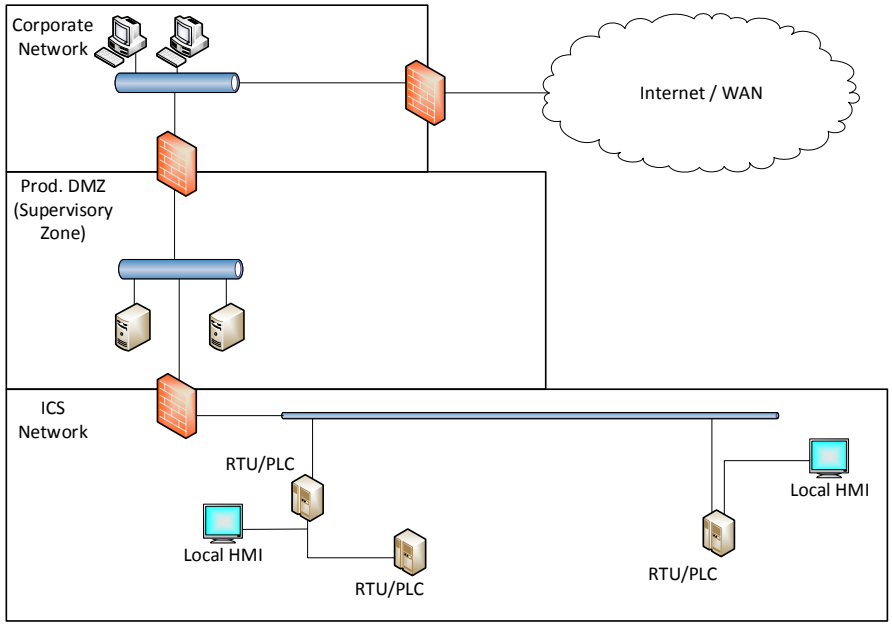

Fig 2. Simplified example network structure of industrial site.

The office network contains all the normal enterprise ICT equipment such as personal computers, corporate WLAN and devices attached to it, Internet connection etc. From the viewpoint of the ICS network and its demands, the office network can be viewed as a hostile environment, with office employees free to act as they please within the boundaries introduced by corporate policies. The office space might also be easier to gain physical access to, when compared with the production environment. (Dzung *et al.* 2005)

ICS networks represent an example of restricted network environment where the approach of machine learning for anomaly detection is feasible. Other restricted networks that exhibit similar attributes are conceivably just as suitable for the deployment of tools and methods leveraging this approach. The nature and scope of the data flowing in the ICS network must remain stable and controllable, and if the addressability and connectivity are used to gain unauthorized access, it must be registered.

When comparing open an restricted networks, we therefore use an ICS network as an example representing the more general category of restricted networks. For a

counter example, an open network, a network of a medium sized university is used. The students are able to access the network from both the university locations as well as their apartments. The network policies are open, and the activity is minimally monitored or moderated.

The restricted or closed nature of ICS networks is currently undergoing a significant change. There is also ongoing research concerning having the SCADA functions to reside in a cloud as presented by Sutic & Atlagic (2013). The major changes in connectivity as embodied by IoT is also causing issues by introducing significant changes to the architectures and configuration of the ICS networks. Mattern & Floerkemeier (2010) note two technical developments of the IoT which are especially important to both the development and to the security of the current and future ICS systems and networks: addressability and communication and cooperation. Both of the developments have security implications. Addressability, defined by Mattern & Floerkemeier (2010) as "Within an Internet of Things, objects can be located and addressed via discovery, look-up or name services, and hence remotely interrogated or configured." has especially far-reaching consequences. Tools such as the on-line search engine Shodan (Shodan, accessed 26/3/2014) can even now be used to search for ICS systems visible to the Internet, with the relevant exploits being served by parties such as the Basecamp Project (Basecamp, accessed 26/3/2014) which also provides penetration testing software Metasploit (Metasploit, accessed 26/3/2014) modules.

The increased visibility to the outside world is not a hindrance for using anomaly detection based on machine learning in these environments, it only highlights the importance of deploying security measures. The visibility and communication with the WAN has increased ICS systems' susceptibility and exposure to potential cyber attack over the various open interfaces to WAN or office LAN. (Nicholson *et al.* 2012)

Table 1 includes the definitions used for the isolated, restricted and open networks in this work.

For the Printocent network used in work presented in papers IV, VI, VII, the static nature, determinism and limited connectivities were all observed in the operational state. Papers IV, VI and VII discuss these issues and present our investigations. Possibly in large part due to the Printocent facility still experiencing further research and deployment activities, fluctuations were still present.

2.4.1 Static nature

From a dynamism viewpoint, a typical ICS network operates on a very different basis than other more open networks, which are open for a varying number of possibly even unauthenticated users. The university network used as an example previously, is bound to have multiple new devices connected each day, old connections removed and new types of protocols appearing. Such a network is a very dynamic one, the users might also be bridging the network to other external networks and connection routing equipment without breaking any rules.

For an ICS environments no such sudden unplanned changes should be allowed. All changes to the environment should be documented, meticulously tested before deployment and eventually deployed in a controlled manner. In a properly deployed and maintained ICS network environment, dynamism and unplanned changes in the behavior should be very limited (Knapp 2011).

The tight control of any changes to the network or its peripherals result in an environment with minimal unintentional changes. It is still important to note that non-intentional changes are always bound to occur, due to accidents, errors in human judgment, etc.

2.4.2 Predictability

In this thesis, determinism and predictability in network or network traffic are used interchangeably and defined as a status where the state of the network or the traffic can be predicted to a high degree in given boundaries. Claiming that a network is deterministic is therefore not to be interpreted as claiming that the future network traffic and network state can be accurately calculated for some t_{0+n} when t_0 is known.

Using the same university network as an example, the determinism of daily communications when compared to ICS network is definitely low. Even while the university network might exhibit some determinism at some intervals, the various attributes are prone to changing without forewarning, without the change representing any type of actual anomaly worthy of closer look. A good example would be a sudden fad taking hold of a substantial part of the student body, triggering a change of proportion respective to the amount of students affected and the type of the fad in question.

Again, for ICS network, such trends should not occur in any reasonable and benign scenario. Having such would point to severely faulty equipment, configurations or

similar issue that would allow users to act their will upon the environment. By definition, ICS networks should contain only the traffic absolutely necessary for the operations of the automation and data collection and their possible safeguard mechanisms. No unnecessary hardware, software, network traffic or software should be present. However, it should be noted that typically these are present in a ICS environment that has been around possibly for decades and has thus evolved considerably over the years. Even if this is the case, it does not alleviate the issue. All unnecessary systems contribute to the increased size of the attack plane. They also possibly harbor well known vulnerabilities, typically present in legacy systems that run software which is either difficult to maintain, or no longer maintained. (Knapp 2011, Langner 2011a, Hadeli *et al.* 2009)

Therefore, ICS networks behave in a predictable manner to some degree. The degree to which they are predictable depends on the individual network environment. The lack of predictability already points to a need to overhaul the network or its peripherals. The level on which the determinism occurs still might vary, as the processed controlled might be subject to non-deterministic processes that could affect the operations. (Knapp 2011, Langner 2011a, Hadeli *et al.* 2009)

2.4.3 Limited connectivity

Starting again with the hypothetical university campus network, it has a part with very open policy for connectivity and use. Students are able to connect nearly any device to the network, using WLAN access points that might be found in the university public areas, or using their own access points at home. The number of devices or their nature is not limited, barring normal limitations for criminal or unethical activity which are not strictly enforced. Typically, the network would have a more restricted or partially separated network segment wherein the restricted networked resources such as employee printing facilities would be connected in.

For an ICS network, this type of nearly unlimited connectivity is impossible if reliability and security concerns play any role for the site in question (Knapp 2011, Langner 2011a). Even the number of HMI's should be strictly controlled, and only the necessary information is available to users. Restrictions such as limiting the devices that may be connected to the network and by whom are typical requirements. No new software or hardware should be installed in the restricted zones or enclaves without explicit approval of persons responsible for the environments operations.

From the anomaly detections perspective, this limited connectivity enables the use

of much stricter threshold values that might be feasible for more open networks. If the site policy strictly prohibits any installation of new devices or connections without prior approval, it is within reason to alert on any new devices and connections.

It is worth noting that the connectivity of ICS networks has steadily increased from the times of isolated networks. This trend is likely to continue, and places considerable burden on proper administration and monitoring of the various systems and networks. As the integration between the ICS networks, office networks and similar environments increase, they begin to share their vulnerabilities. (Nicholson *et al.* 2012, Alcaraz *et al.* 2013)

3 Machine learning and anomaly detection

The process to select a suitable machine learning approach to be used in modeling network traffic is a well-studied issue. Numerous different algorithms have been proposed to be used in this context and some have been implemented on some platform and tested for their performance. The general approaches of supervised, unsupervised, reinforcement and deep learning are discussed in their respective sections in this chapter. However, we begin with a look at the different machine learning algorithms as used or investigated for use in network anomaly detection.

Using machine learning for anomaly detection in settings other than data networks is a very diverse field. In this work, we focus only on the machine learning based anomaly detection in the context of data networks, and restricted networks such as ICS networks in particular. This should be kept in mind when we discuss the subject in this thesis.

Machine learning anomaly detection faces significant issues when brought from the closed environments into a more open one Sommer & Paxson (2010). However, the issues causing the difficulties itemized in Sommer & Paxson (2010) are either less severe or mostly absent in ICS or other restricted network environments, as explained in Section 2.4.

Different machine learning approaches have been investigated for use in anomaly detection for network traffic. Examples of these include the research work published in the work discussed below.

García-Teodoro *et al.* (2009) classify machine learning based anomaly detection, or anomaly based network intrusion detection as defined by the authors, into six subtypes. The subtypes are depicted in List 1. The authors note that machine learning approaches are very adaptable, flexible and capture interdependencies well. As a negative side, they note that the resource consumption is high and the approaches typically rely strongly on the assumed accepted behavior.

Ye *et al.* (2004) investigate the robustness of a Markov-chain model based system for intrusion detection purposes. The results of the paper show that the Markov-chain approach suffers from deteriorating performance when exposed to noise. This sensitivity to noise is in line with one of the challenges facing machine learning intrusion detection presented by Sommer & Paxson (2010). Based on the results of the paper, the Markov-chain model seems not particularly well suited to general network intrusion detection.

List 1. Machine learning anomaly detection subtypes as defined in García-Teodoro *et al.* (2009).

1. Bayesian networks
2. Markov models
3. Artificial neural networks
4. Fuzzy logic
5. Genetic algorithms
6. Clustering and outlier detection

However, this does not mean that it might not still be usable for environments in which there is little or no extra noise present.

Restricted Boltzmann Machines (RBM), or more specifically, Discriminative RBM (DRBM), is investigated by Fiore *et al.* (2013) for anomaly detection. The results provided point to a promising approach. They use a similar non-labeled approach as is implemented by the MBM, no previous information on anomalous traffic is available. Fiore *et al.* (2013) note that the main issues related to the performance of the DRBM classifier as the "randomness and burstiness of the traffic behavior." combined with the lack of training data representing the actual normal traffic. These issues largely do not affect, or should not, affect ICS networks (Knapp 2011).

Parikh & Chen (2008) introduce a system based on data fusion and implementing a "Hierarchical ensemble of classifiers". The authors claim that the performance of their new algorithm, called dLEARNIN, surpassed that of the other reported results for a work which used comparable pattern recognition algorithms. The algorithm combined data from several sources. Authors used the DARPA 1999 dataset for comparing the results to other approaches.

Mabu *et al.* (2011) introduce a model for network intrusion detection. The model makes use of genetic network programming. The model includes a system for both misuse and anomaly detection. The authors report good performance for anomaly detection use case, with high detection rate and low, if not optimal, false positive rate. They use genetic network programming (GNP), an evolutionary form of genetic algorithms approach (Hirasawa *et al.* 2001), to extract rules from normal connections. These rules are then used as a model for normal traffic for which classifiers are based upon.

Adler *et al.* (2013) use both k-means clustering and support vector machine (SVM) with a set of features with considerable overlap with the features selected and used in the MBM system and investigated in articles I, II, III,IV and VII. The features include,

e.g. address 4-tuples, including IP and port values and durations of the connections, or flow. The paper by Adler *et al.* (2013) also includes interesting positive commentary from US Air Force personnel in regards of automated machine learning systems, as described in the paper and also represented by the MBM system presented in Chapter 4.

Shon *et al.* (2005) introduce and demonstrate a framework aimed at anomaly detection and implementing both genetic algorithm (GA) approach and SVM. The authors proceed to test the approach with the DARPA IDS 1999 dataset. They also test the system against both Snort (Snort, accessed 1/12/2013) and Bro (Paxson 1999). It is unclear whether Bro was used in the signature matching capability or in the native event-driven mode. However, the event-driven mode might be suggested by the fact that Bro outperformed Snort in the tests. The GA and SVM framework and approach introduced by the authors is also demonstrated to produce comparable results. The work by the same authors in (Shon & Moon 2007) demonstrates a hybrid system consisting of self-organizing map combined with SVM and GA approaches. The authors call this SVM combined with SOM, GA and other additional techniques an "Enhanced SVM", it is clearly an evolutionary step from their earlier work in (Shon *et al.* 2005). They then proceed to compare the performance of their enhanced SVM hybrid system. The 1999 DARPA IDS dataset is used and the comparison is again done against Snort and Bro as in (Shon *et al.* 2005). In the paper (Shon *et al.* 2005), the authors use SOM to pre-process the network data for SVM learning, relying on its clustering functionality and then proceeding to label them according to their nature.

Tsai *et al.* (2009) classify machine learning techniques for intrusion detection into four different classes: pattern classification, single classifiers, hybrid classifiers and ensemble classifiers, depicted in Table 3. They use a definition of anomaly detection as a sub-set of intrusion detection. The classification is unclear at some points, e.g. why SOM is classified separately from artificial neural networks. According to Haykin (2009), Kohonen *et al.* (2001), SOM is considered belonging to the ANN family of machine learning algorithms. The definition is also different compared to the one provided by García-Teodoro *et al.* (2009). In a later paper, Tsai & Lin (2010) introduce a hybrid system leveraging machine learning for intrusion detection purposes.

Early on in our approach, we decided to move forward with the approach of using unsupervised learning to model the patterns inherent in the network traffic. This approach requires no prior labeling of input data, but relies on the information inherent in the input itself. The approach lends itself well for discovering patterns in the input data, and that is in fact one of the common usages for algorithms of this type. (Murphy

Table 3. Machine learning techniques for intrusion detection classified as single classifiers by Tsai *et al.* (2009).

Technique	Elaboration and examples
Support vector machines	According to Shon & Moon (2007), one of the best ML algorithms for classifying abnormal behavior. It is based on statistical learning theory (Vapnik 1999).
K-nearest neighbor	A Straightforward yet performance-wise competitive pattern recognition algorithm (Manocha & Girolami 2007). It has been proposed for AD system use, e.g. by Om & Kundu (2012), where it functions as a part of a hybrid system.
Artificial neural networks	ANN family consists of multiple different algorithms (Haykin 2009). ANN approaches have been investigated and used for anomaly detection, e.g. RBM (Fiore *et al.* 2013), (Lee & Heinbuch 2001).
Self-organizing map	A widely used ANN algorithm Kohonen *et al.* (2001). Network anomaly and intrusion detection implementations exist, e.g. (Ramadas *et al.* 2003), (Kayacik *et al.* 2007) and the MBM depicted in Section 4.
Decision trees	Used in approaches for anomaly detection, e.g. by Sindhu *et al.* (2012) and Muniyandi *et al.* (2012) in combination with other algorithms.
Genetic algorithms	Used for anomaly detection, e.g. by Shon & Moon (2007) and also in its evolutionary forms, such as the GNP (Hirasawa *et al.* 2001), by e.g. Mabu *et al.* (2011).
Fuzzy logic	Anomaly detection solutions exist, e.g. by Linda *et al.* (2011) who target AD in CI environments. Fuzzy component is also present in other approaches, e.g. (Mabu *et al.* 2011)
Naive Bayesian networks	Approaches such as by Tylman (2008) who present a Bayesian network derived solution for AD based on the Snort and by Om & Kundu (2012) where naive Bayesian network is a part of a hybrid solution.

2012, Russell & Norvig 2010, Haykin 2009)

At the heart of the problem, lies the task of selecting the most useful features to be included in the implementation of the machine learning algorithm eventually used. The selection of features is also dependent on the algorithm used and vice versa. The algorithm should be selected to best fit the features that optimally represent the data important to the problem at hand, but this also creates demands for the dataset, such as a requirement for labeled input for supervised algorithms. This creates an interesting problem, as we are not initially sure what features we would like to use or what algorithm, and we might end up as needlessly constrained by either selection done too early in the process. (Russell & Norvig 2010, Bengio *et al.* 2013)

The aim of our initial studies was to discover a suitable algorithm for discovering outliers from the clusters formed by the algorithm in the learning phase. One of the suitable algorithms was the SOM (Kohonen *et al.* 2001). The algorithm was chosen for our initial test implementation and is discussed in more detail in Section 3.4.1.

Significant work has been done on using machine learning algorithms for network security monitoring and anomaly detection. The SOM algorithm has been also been investigated for this particular purpose, such as in work presented by Ramadas *et al.* (2003) and Kayacik *et al.* (2007). Research presented in Sarasamma *et al.* (2005) investigates a multilevel hierarchical approach and comparison to a single level approach, with the single level approach being the initial ML algorithm implemented for MBM. The work presented in the paper by Lee & Heinbuch (2001) investigates the training of neural network anomaly detectors for intrusion detection purposes and also brings up the challenge of the variability present in most networks. Work in Hu *et al.* (2004) investigates the usage of fuzzy SOM's in comparison the classical implementation.

Using a neural network approach was judged feasible based on the literature review, e.g. (Ramadas *et al.* 2003) and (Kayacik *et al.* 2007). Neural network approaches for network security monitoring purposes, particularly intrusion detection, have also been investigated early on, such as by Debar *et al.* (1992).

Other algorithms than SOM could have been selected as well. SOM provided a well tested starting point for our investigations to see if an artificial intelligence algorithm of neural network type would adapt well to the modeling of restricted network environment traffic and state.

3.1 Different machine learning approaches

In this section, various machine learning approaches are discussed in a more general level. This section does not provide lists or descriptions of the various individual algorithms that have been or might be used for anomaly detection in a communication or control network. The aim is to provide an overview of the more general approaches of machine learning. Few algorithms are discussed in more detail to provide the reader with a deeper insight into the approach discussed in the corresponding section and its general applicability for the network anomaly detection.

The hierarchical structure of this section is not strict. The supervised and unsupervised learning are very basic concepts of machine learning. Reinforcement learning can be interpreted as a third category along with supervised and unsupervised learning Murphy (2012) even though it is more rarely used, and we will only give it a cursory discussion in Section 3.1.3. Semi-supervised learning means an intermediary solution between supervised and unsupervised learning, and is not discussed in here. Deep learning is of a higher abstraction layer, using the more basic approaches itself. Deep learning is presented here because it is seen as a relevant approach for the topic of this chapter.

It is noteworthy that a machine learning based AD approach is not confined to a single classification. Combining methods from unsupervised, supervised and reinforcement learning can be used.

3.1.1 Supervised learning

Supervised learning denotes a process where a system is taught using labeled data. Labeling input data is an effort intensive process for any larger data set. (Murphy 2012, Russell & Norvig 2010, Haykin 2009)

Given a set of labeled data, there are several possible algorithms for use, also for anomaly detection in network environments. One such algorithm is support vector machine (SVM), for which an anomaly detection implementation is presented in Mukkamala *et al.* (2002) where it is also compared to an ANN approach.

Naive Bayesian classifiers also represent the supervised learning algorithm type of machine learning. An example solution from the literature for anomaly detection using this algorithm for a hybrid solution is presented in Om & Kundu (2012).

A challenge for supervised learning in the network anomaly detection context

is the need for a corpus of labeled input data. To allow a system using this type of learning approach to build a model of a new network, the network traffic input would need to be labeled. This is a time and resource consuming effort which also requires considerable expertise from the teams or individuals responsible for it. Finding a party with enough expertise to commit to such an undertaking is also not trivial for specialized and very heterogeneous networks. The network traffic of ICS networks is typically deterministic Hadeli *et al.* (2009), as previously discussed. However, due to the multitude of equipment vendors and long time frames of updates and configuration changes, they are by no means homogeneous either, as single networks or between different networks on different sites.

Supervised learning is very accurate when properly trained, but the training and labeling of the data represent a challenge.

3.1.2 Unsupervised learning

Unsupervised learning process does not require prior labeling of the input data. The algorithms which use this type of learning rely on the information inherent in the data itself and typically are well suited for classification or clustering tasks. The human learning also largely relies on unsupervised learning. Other learning methods are also in use, the ways biological creatures of higher order learn is a combination of multiple styles of learning. Unsupervised learning, and especially ANN algorithms, are almost always modeled after a distinct biological learning phenomenon. (Murphy 2012, Russell & Norvig 2010, Haykin 2009)

Given that ICS networks are typically very heterogeneous but exhibit deterministic traits and are restricted in nature, unsupervised learning is well suited for constructing a model of them. Using a suitable unsupervised learning algorithm, the network traffic and states can be mapped to a model and this model then used to detect deviations or by mapping certain clusters to represent anomalies or attacks.

There are several algorithms that represent this class, such as the RBM which has also been used for anomaly detection in network traffic in a work published in paper by Fiore *et al.* (2013). However, in the paper, usage of RBM is termed as semi-supervised as the training data is assumed to be anomaly free.

Self-organizing map algorithm as used in Kayacik *et al.* (2007) and Ramadas *et al.* (2003) also represents unsupervised learning. The typical usage could also be noted as semi-supervised allowing for the notation used in Fiore *et al.* (2013).

3.1.3 Reinforcement learning

Reinforcement learning is based on the concept of a system receiving positive or negative feedback based on its actions, or output Murphy (2012).

Such learning methods are described, for example, in the work presented in papers by Barto *et al.* (1983) and Prokhorov & Wunsch (1997). Particularly the paper by Barto *et al.* (1983) documents very seminal work in this field. As is the case with artificial neural networks, there are biological analogues to be found in the animal kingdom for reinforcement learning (Haykin 2009, Barto *et al.* 1983).

Reinforcement learning with the negative and positive feedback loop could well provide additional capabilities for a system such as the one described in Chapter 4. One example could be that the output that is judged a false negative could be used to provide feedback for the system to take this information into account in further processing.

3.1.4 Deep learning

The trend of Deep learning is a relatively new area in the field of machine learning. A basic assumption for deep learning is that a multi-layered architecture can be used to store and process information at varying levels of abstraction. The deeper parts of the network process very low level information and the level of abstraction is increased as the higher layers are approached. (Bengio *et al.* 2013, Bengio 2009, Murphy 2012)

Deep learning is a concept, which can be approached in different manners. The book by Murphy (2012) lists some possibilities such as deep directed networks, deep Boltzmann machines, deep belief networks, greedy layer-wise learning of DBNs and various deep neural network approaches.

As presented in the paper by Bengio *et al.* (2013) approaches leveraging deep learning have excelled in a number of tasks. The tasks are listed in List 2.

From List 2, we can see that some of the fields deep learning has excelled in do overlap with domains where SOM has been applied with good results. Speech recognition has been a traditional example for SOM usage, with the phonetic typewriter capable of producing text from speech input as a classical example (Kohonen *et al.* 2001).

One method to construct a classifier using a deep learning approach is to learn the features, or hierarchy of them one layer at a time, each higher layer learning the representation of the output of the layer below it. After the learning, the architecture can

List 2. Examples of deep learning application domains with good performance.

1. Natural language processing
2. Recognition of objects
3. Recognition of speech
4. Signal processing
5. Multitask learning
6. Transfer learning
7. Domain adaptation

be used to build a supervised predictor. (Bengio *et al.* 2013)

The use of unsupervised predictor for a deep learning architecture is not as well developed as the use of supervised predictor (Bengio *et al.* 2013). This creates some issues for the use of this approach for the purely unsupervised-learning based anomaly detection approach presented in this thesis.

Despite the lack of proper unsupervised predictor architecture, deep learning is a field moving very rapidly forward. It appears to be feasible for use in anomaly detection in data networks as well. However, this requires more investigation into the matter and advances in the unsupervised predictor issue if purely unsupervised learning based system is desired. As of January 2014, no information on deep learning based anomaly detection systems for data networks was discovered through a literature review. This points to the fact that no such system exists, or least no-one is yet at a stage where they would be willing to publish their findings, excluding possible proprietary implementations.

3.2 Benefits

Operationally, a functional machine learning based network security monitoring system geared towards intrusion detection has been a lucrative and elusive topic in the intrusion detection research. In the ideal setting of closed laboratory environments, good results have been obtained. (Sommer & Paxson 2010, Kayacik *et al.* 2007)

In Adler *et al.* (2013), expert commentary by unnamed US Air Force stakeholders also places considerable interest in the applicability of automated tools for security purposes. If the challenges can be overcome, the good laboratory results point to the possibility of good results in operational settings as well.

The benefits of having a system capable of learning its network environment and

directing its detection based on this automatic process are fairly clear. A properly configured and near-optimally operating system could possibly be used to detect several types of zero-day attacks and other difficult targets. The reduced amount of time required for manual configuration would also yield significant savings.

The benefits are listed in Table 4.

Table 4. Examples of the benefits of ML based anomaly detection where feasible.

Benefit	Rationale
Cost savings	Increased automation and the decreased requirements for manual tuning of the system.
Zero-day detection	Possibility for detecting previously unseen and totally unknown threats when the threat realization causes fluctuations in the features used for monitoring.
Advanced persistent threat detection	Possibility for detecting even highly tailored attacks not found in the wild as might cause notable fluctuations.
Cyber-attack detection	Detection of fluctuations and abnormal traffic patterns caused by cyber-attacks.
Intruder activity detection	Detection of fluctuations and abnormal traffic patterns caused by otherwise permitted actions by intruders that have already penetrated the network.

3.3 Challenges

In the paper by Sommer & Paxson (2010), Sommer *et al.* discuss the applicability of machine learning for network intrusion detection and difficulties arising from bringing the system out of the laboratory and into the open. This work has been instrumental for our work in mapping the difficulties of using machine learning for intrusion or anomaly detection. In paper I, we also mapped some of the challenges pertaining to the use of machine learning in ICS environments, using the paper by Sommer & Paxson (2010) as an important source.

When using machine learning approach based on modeling, the known state of the traffic it is also difficult to provide the user with information concerning the nature of the detected anomaly. As the model has been built to include the known benign traffic, it

Table 5. Challenges for using and validating ML for network anomaly detection (Sommer & Paxson 2010).

Challenge	Elaboration
Outlier detection	ML algorithms are typically geared towards finding a logically similar object, not those which are dissimilar.
Difficult evaluation	Lack of data for learning, reluctance of the network operators to provide data for researchers due to security and privacy concerns.
Loss of logical traceability	Many machine learning algorithms, such as SOM, do not provide a clear line of deduction.
High number of f_p	Sub-optimal operational environment, such as too exposed an office network.
High cost of errors	Both false positives and false negatives are potentially very expensive, in either terms of human labor or damages, respectively.
Diversity of traffic	Establishing base state of network can be difficult. When the network is constantly changing, establishing any kind of a base line is very difficult. Even if it is possible to find some very low level base state, the information gained from comparing the monitored state to that state is limited.
Semantic gap	The difficulty of providing output that can be used to mount concrete actions and decisions.

would be possible to annotate the model to provide the user information about how the new traffic corresponds to the traffic earlier seen. However, as the model contains no patterns based on attacks or other malicious content, it does not directly provide information on these, even if detected as anomalies. Other methods should be employed to provide the user or administrator with extra information. Other approach would be to include malignant traffic patterns in the model created by the machine learning, and to differentiate between malignant and benign traffic through annotating them accordingly or through other means.

In addition to these issues specific to the network anomaly detection, all the typical machine learning issues continue to pose a challenge.

The main challenges to network intrusion detection by machine learning are listed in Table 5 as defined by Sommer & Paxson (2010).

3.4 Applicability in ICS networks

Works describing attempts at leveraging machine learning approaches for anomaly detection in data networks are rather numerous in published literature. The general applicability of the approach to more open networks is not discussed in depth in our work. Good presentation of the open network issues can be found in the paper by Sommer & Paxson (2010), which we also used as a basis for our rationale on why the issues are not that serious in more restricted environments.

Some of the published work, such as that presented in the paper by Hadeli *et al.* (2009) which concerns leveraging the deterministic traffic patterns exhibited by ICS networks for anomaly detection directly converges with our own work and backs up the basic assumption.

Work presented by Linda *et al.* (2009) describes a neural network based intrusion detection methods aimed for critical infrastructure settings with good test results. Linda *et al.* (2011) also introduce a fuzzy logic based approach for anomaly detection with good results. Linda *et al.* (2012) introduce a system for anomaly detection architecture aimed at ICS environments.

Leveraging of the attributes presented in the Section 2.4 for ICS networks is the core of our claim that machine learning can be used. We find the same rationale behind the work presented in several recent research articles. For example, in the recent work presented by Goldenberg & Wool (2013), Morris *et al.* (2012, 2013), Valdes & Cheung (2009), Hadeli *et al.* (2009), Linda *et al.* (2009), Yang *et al.* (2006) authors make note and use of some or all of the attributes listed in 2.4.

Leveraging the static nature of the ICS networks has been a basis for significant research, the static nature, or at least decreased dynamism and the predictability or determinism are parts of the same whole. The work presented in papers by Linda *et al.* (2009, 2011, 2012), Hadeli *et al.* (2009) converges with the research of this thesis and strongly supports the feasibility of using machine learning for anomaly detection solutions in ICS environments.

Leveraging the predictability and decreased dynamism of ICS networks we avoid for the most part the challenges of diversity and variability Lee & Heinbuch (2001) found in the more open network environments. In the work presented in the paper by Yang *et al.* (2006) no machine learning approach is used, but the authors make note that certain anomaly detection approaches as documented by them seem "especially applicable to SCADA system security which are characterized by routine and repetitious activities."

This type of activity can be used to build a model using machine learning algorithms in a relatively straightforward manner.

In an optimal situation, ICS networks would not be connected to the outside world, except through a tightly controlled DMZ, and no direct connections between even the corporate network and ICS networks should be allowed. This again allows us to gather information and be aware of what types of connections should exist at a given time and state of the ICS environment. Deviations from previously seen states typically point to changes in the environment, malfunctions, or even intrusions, as presented in papers, e.g. in VII.

3.4.1 Selected algorithm for anomaly detection PoC

SOM is an algorithm well suited for a good number of tasks and it is widely used for solving a variety of real-life problems. It is an unsupervised learning algorithm that maps multi-dimensional data into a selected dimensionality map. A two dimensional map is a classical one, but others are used for a number of applications. Kohonen model presented by Kohonen (1982) has received significantly more literature coverage over Willshaw-von der Marlsburg's model by Willshaw & Von Der Malsburg (1976) and the Kohonen model was chosen for PoC implementation.

The selection of SOM as the first algorithm for the anomaly detection module was not immediately clear. However, SOM had been studied before for use in this task and successfully implemented and tested by several parties, as for example presented in the papers by Ramadas *et al.* (2003), Kayacik *et al.* (2003, 2007). Usage of SOM for anomaly detection in general is therefore not a novel approach by any means, it is a well tested one which shows promise for this particular context with other machine learning approaches. Therefore, it was well suited as the initial algorithm to be implemented using Bro NSM scripts.

Kayacik *et al.* (2007) apparently uses Bro output logs to generate input data for their hierarchical SOM implemented using SOM-PAK Kohonen *et al.* (1996) and some of the features used are the same as those used by the MBM. Namely, the following features overlap: duration of connection, bytes sent by originator and receiver of the connection. Kayacik *et al.* (2007) investigate the feasibility of a hierarchical SOM usage in a general intrusion detection context. The information provided in the publication proved very useful, even while unfortunately discovered by the author of this thesis at a comparatively late stage. The results of Kayacik *et al.* (2007) are promising for using

List 3. SOM Properties, (Kohonen *et al.* 2001, Haykin 2009).

1. Input space approximation
2. Topological ordering
3. Density matching
4. Feature selection

the SOM for anomaly detection in ICS setting.

Ramadas *et al.* (2003) use a more separate IDS system with an SOM extension. The SOM extension of the system presented in the paper also makes use of the SOM-PAK as did Kayacik *et al.* (2007).

The basic SOM algorithm is not a very complex one and is well suited for use for this type of an problem. Basic SOM features are presented in List 3 Kohonen *et al.* (2001), Haykin (2009). If normalization of input data is done properly and the selected and implemented features are adequate, it is able to cluster the network traffic in a way which allows for detection of traffic not included in the model. The logic by which the traffic is categorized as not included can be implemented in several ways, one such way is simply to compute the Euclidean distance, decide on an threshold value and decide that anything far enough by the Euclidean distance is not included.

SOM algorithm implementation for anomaly detection does suffer from a number of issues when facing a network environment. Most of these issues equally apply to a number of other situations as well.

Firstly, it is prone to get stuck in the way of the activity converging to one or few neurons with rest of the lattice remaining in a state where they are not excited at all by input. This creates a situation where one solution would be to just start the whole learning process anew. It is also noteworthy that the mathematical proof for the convergence of SOM lattices has been elusive. No accepted proof for the convergence has been put forward as of the time of this writing to the knowledge of the author. Convergence issues are discussed in Erwin *et al.* (1992).

The incomplete proof of convergence remains a hindrance for SOM applicability as it is therefore lacking a complete mathematical representation. This introduces additional uncertainty to the process of training a system relying on SOM.

SOM also requires a considerable amount of training data to be used when training the algorithm Kohonen (1982). Especially in situations such as the network anomaly detection, this has implications against development of effective feature sets and systems, as network traces can be hard to obtain.

Other algorithm might have been chosen as well, there are several machine learning algorithms that are suited for this type of an approach. The SOM was selected for our particular test implementation and validation presented in paper VII to show that the algorithm works adequately.

The basic and original SOM algorithm is explained in detail in the book by Kohonen *et al.* (2001) and included here for completeness and explanations particular to the approach used in the PoC system implementation. The algorithm works in an incremental fashion, processing a new input feature vector at each selected interval increment of time t or other value.

The basic learning process is depicted in equation 1. In our approach, the initial values of m_i are initialized to random values in the range $[0,1]$. The time t is replaced by each step representing the next connection and the actual time difference varies. The $||x_t - m_i(t)||$ denotes the Euclidean distance between the input vector and the weight vector. Other distance heuristics could be used as well, but in our implementation we, also use the Euclidean distance.

$$m_i(t+1) = m_i + h_{ci}(t)[x(t) - m_i(t)] \tag{1}$$

The neighborhood function of equation 1, h_{ci} is classically the Gaussian function depicted in equation 2.

$$h_{ci}(t) = \alpha * \varepsilon^{\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right)} \tag{2}$$

In this neighborhood function $||r_c - r_i||$ denotes the Euclidean distance between the location vectors of nodes c and i depicted in equation 3 for two dimensional Cartesian coordinates.

$$d_{euc}(i,j) = \sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)} \tag{3}$$

In the PoC implementation, we use the Chebyshev distance between the location vectors, given in 4 resulting the neighborhood function of 5 where θ and α_{mod} can also be modified by the users.

$$d_{che}(i,j) = max(|x_i - x_j|, |y_i - y_j|) \tag{4}$$

$$h_{ci}(t) = \alpha_{mod} * \varepsilon \left(-\frac{(d_{che}(c,i))^2}{\theta(t)} \right) \tag{5}$$

For every input vector, we look for the closest weight vector, given by equation 6 where m_c denotes the closest weight vector.

$$||x - m_c|| = \min_i ||x - m_i|| \tag{6}$$

For each node in the neighborhood of the winning node m_c, we compute the learning rate factor $\alpha(t)$ which decreases as the Chebyshev distance increases from the m_c when using a function similar to the Gaussian kernel depicted in equation 2. The $\alpha(t)$ also typically decreases over time. In our case, the $\alpha(t)$ decreases according to the epochs passed and connections seen.

3.4.2 Feature selection

The selection of the features, or the data representation, for a machine learning algorithm implementation to be used for any practical purposes is an important issue Bengio *et al.* (2013). The features need to be selected in a way that the attributes which are to be monitored are represented by that combination of features and the resulting machine learning model is therefore in principle capable of producing the desired output. Poor selection of features will result in a machine learning system which is not capable to provide the desired functionality in an optimal manner.

As a part of the feature selection phase, the normalization or lack of it needs to be addressed. The input values need to be parsed and formatted in a way which incurs no or minimal information loss in the process.

List 4. Currently implemented features for SOM extension.

1. The number of live TCP connections at the moment of connection termination
2. The number of live UDP connections at the moment of connection termination
3. The number of live ICMP connections at the moment of connection termination
4. Duration of a connection that terminated
5. Overall network fragments pending reassembly by Bro
6. The amount of data (bytes) sent by connection responder
7. The amount of data (bytes) sent by connection originator
8. Number of packets sent by the connection responder
9. Number of packets sent by the connection originator

In our work with the SOM algorithm, all features are normalized to values between 0 and 1. The normalization is done by using earlier maximum seen for that particular value with consideration given to things such as the used transport protocol.

The currently implemented features for the MBM system can be found in List 4. Since the features that have high variance due to the transport protocol in question are normalized by the corresponding maximum seen, the features concerning duration, data amounts and packets sent that are itemized in the table could be further divided into three sub-features for TCP, UDP and ICMP. More information on the features can be found in papers II, III, IV and VII, as well as the description of system implementation in Chapter 4. Similar, or similar in part, feature sets for SOM have been investigated by other parties as well, e.g. by Kayacik *et al.* (2007) Ramadas *et al.* (2003) and Linda *et al.* (2011, 2012).

Several other features could be used for anomaly detection using SOM in the setting of ICS networks. List 4 is not an exhaustive list of features investigated or currently under investigation for use, it merely documents the features available for constructing the feature vector at the time of a single connection termination for MBM.

The approach for triggering SOM activity only during connection termination is also sub-optimal, as this approach will be one step behind any malicious activity. Triggering of activity and the selection of suitable features and possibly algorithms for other events is currently under investigation.

4 MBM system implementation

The machine learning based anomaly detection system is built as an extension of the Bro NSM Paxson (1999) and implemented with the network domain specific scripting language of the system. The extension system has been introduced initially in paper IV and further elaborated, validated and tested in paper VII. In this Chapter, an overview of the system status is briefly presented. The base functionality of the MBM module is to provide a modular framework for *event-driven machine learning anomaly detection* to be used in ICS network settings. The SOM algorithm and the handling of `connection_state_remove` event currently serve as a PoC implementation demonstrating that it is indeed feasible to construct such a system using Bro NSM.

The formulation of EMLAD concept is briefly discussed in papers IV and VII. In this approach, a machine learning algorithm is tied to a context of a single event. This event can be either low or high level, and the features used by the algorithm need to be defined per the event handled. The same algorithm can be used for handling of several different events, but the constructed models are structured per event. Each event can have multiple machine learning algorithms handling its processing. Each algorithm will need as many instances as there are events to be handled by it. Each such instance will generate and store its own model of the historical data as per the specific algorithm in use. This is still at the concept stage, with much implementation still under way.

The current implementation consists of a number Bro script files. Two of the scripts, `SOM.bro` and `SOM_config.bro` contain the SOM related functionality. The scripts `ml-core.bro` and `ml-core_config.bro` contain the general module functionality and the logic for selecting machine learning algorithms for different events. The other files contain integration scripts for other systems and redefined parameters for passing as command line arguments.

The paper IV which includes the initial introduction of the system includes diagrams on drawings on the system status at the point of the papers' publication, and there are a number of changes. One of such changes is that the base installation of Bro does not need to be changed. The extension system can be invoked by passing the relevant script files and parameters as additional scripts. In the initial implementation, there were hooks written into the Bro base installation, which is now no longer required, improving usability and making for an easier installation. As the core Bro installation needs no

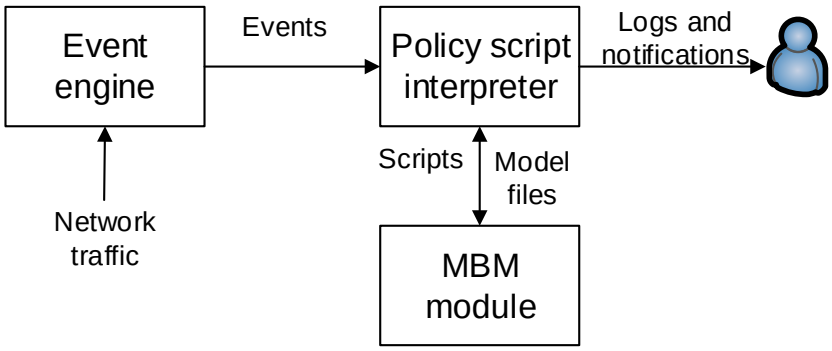

Fig 3. The relation of the MBM script module to Bro.

changes, the MBM can be deployed alongside any Bro deployment to complement any other functionality used. Figure 3 illustrates the positioning of the MBM as simply a script module for the normal Bro system deployment.

Currently, only a SOM algorithm has been implemented for the machine learning extension system. The basic algorithm and modifications used were introduced in Section 3.4.1. As the Bro scripting language has not been intended for implementing machine learng algorithms, some peculiarities are present in the code, such as a self-recursive way SOM lattices are initialized which creates some issues. The issues include items such as initialization failures in the form of segmentation faults given too large initial lattice size. If larger lattices would be needed, initializing them using a shell script would be a simple solution.

The system does not currently include any visualization mechanisms for the created SOM lattices. The visualization of the SOM lattice saved in the text file should be fairly straightforward if such a functionality would be needed some time in the future. This could be achieved with some parsing mechanism coupled with an existing SOM visualization tool.

The processes for either of the modes, detection or learning, are depicted in Figure 4. In the figure, it is clearly marked, which of the processes are handled by the extension module, and which are the normal Bro functionality. Bro receives its input through either monitoring a live network interface, or by reading from a packet capture file.

Important work on the Bro and work concerning leveraging the tool also includes

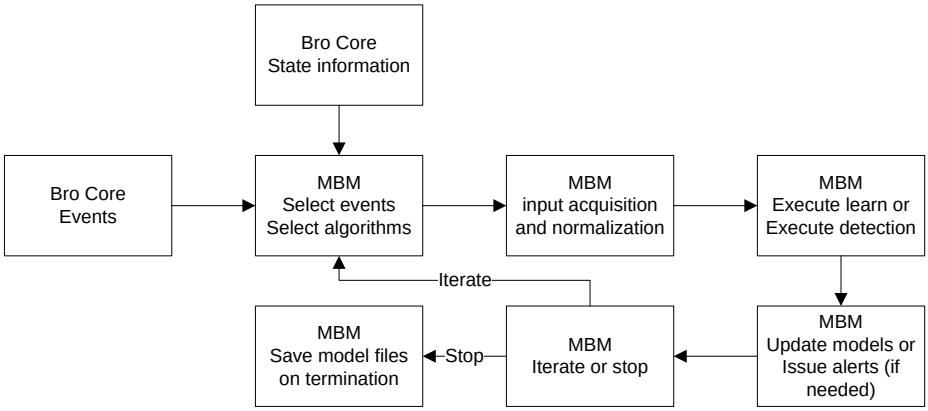

Fig 4. The general operation processes for either detection or learning modes.

work such as presented by the researchers in the following papers: (Vallentin *et al.* 2007, Weaver *et al.* 2006, Dreger *et al.* 2008, Gonzalez & Paxson 2006). Bro related information and downloads are available on the projects web site at Bro NSM, (accessed 2/12/2013).

4.1 Implemented features

Currently, the features implemented, as depicted in List 4, include mostly information concerned with the transport protocol level. This issue arises from the fact that as of this writing Bro NSM did not include parsers for protocols specific to ICS environments, with the exception of DNP3 Lin *et al.* (2013) and MODBUS over TCP which became available with Bro NSM version 2.2.

Implementation of features specific to the protocols ub the network environment where the system is to be used would require either that the protocol parsers needed exist, can be written by the users, or that a higher level processing is used. For example, if the protocol being transported over TCP is unknown, we look at the features of the TCP protocol. Similarly, if the transport protocol over IP is unknown, the IP protocol is used. In this work, we limit the study to protocols transmitted over IP.

If parsers are available for the protocols present, the numbers features that can be created for the SOM would be significantly higher than for mere transport protocols. The available contrast and resolution would also be improved, as the protocol specific details can be extracted and used in the feature vectors.

The test traces currently used do not contain any DNP3 traffic, and as of October 2013, there exist no parsers for the Siemens Simatic proprietary protocols most abundant in the traces of Printocent MAXI line. However, should a protocol parser for the Siemens protocol become available, it would provide significant improvement to the accuracy and sensitivity of the system with additional features. It would enable us to create protocol specific features such as the values of the control traffic transmitted.

4.2 Normalization

Normalization in a default set-up is accomplished by dividing the seen information with the maximum seen by the system. For features such as *duration* where the feature value can dramatically vary between different transport protocol, this division is further broken down to the maximums seen for the specific transport protocol in question. The user can easily modify the normalization functions as seen fit.

Normalization dramatically affects the performance of the system. If the *duration* is not normalized by the maximum seen for the corresponding transport protocol, but to the maximum connection duration seen for any connection, poor performance results. This is due to the fact that for example a long TCP connection would result in most of the short UDP connections getting normalized very close to zero. This again would provide poor resolution between the durations of UDP connections, as they would all get near zero values for their weight vectors after being normalized with the large value from the longest TCP connections.

The way the normalization is currently done using the maximum value seen creates a situation where the normalized value of a given input can change over time. This causes some issues during the initial learning period, but after the first epoch, the value used for normalization is the global maximum of the whole training set for the particular input. Therefore, this only affects the initial epoch and causes no further issues down the line. Having the maximum values set by the first run, and not doing any learning would be another way to approach this. That could be accomplished by simply inhibiting the updating of the nodes in the lattice, a quick and simple fix. However, as the issue did not seem to cause any trouble, this modification is currently not implemented.

If the user wishes to add more features to the system, one of the important things to consider is the handling of normalization. Otherwise, a useful feature can be rendered meaningless through poor normalization.

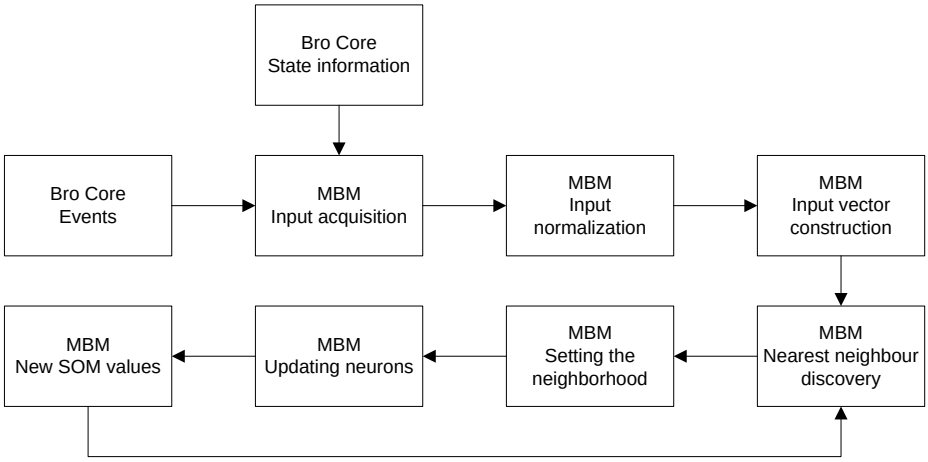

Fig 5. The SOM extension module learning process.

4.3 Learning mode

The very first step in the learning mode is the initialization of a SOM lattice with random values in the range of $[0, 1]$. After this, the user can select the ordering and fine tuning periods by crafting suitable shell scripts that invoke the system with suitable values. Currently, issues with the very first initialization of the SOM lattice have issues with very large lattice initialization. Lattices at around 10K neurons sometimes cause segmentation faults and very large lattices need to be initialized outside of the Bro. The initialization of fresh lattices is done in a very sub-optimal manner, due to some of the features of the scripting language. Writing a companion shell script for the initialization of lattices is definitely something that might be useful in the future. However, as the lattice size grows, so do the requirements for computational resources. For improved performance, an minimal sized lattice that yields accurate enough results should be used. There is a significant change in performance when the lattice size is increased.

The main learning function is presented in listing 4.1 with the learning process depicted in Figure 5 omitting the initialization and termination processes.

Listing 4.1. Learning logic debugging prints removed and lines wrapped for improved readability

```
##################################################
# Functionality for teaching the neural network
#
function SOM_learn(
c: connection, proto: transport_proto): count
{
local SOM_input_vector: vector of double;
local closest_node: vector of count;

# Obtain the input and normalize for input vector
SOM_input_vector = SOM_build_input_vector(c, proto);

# Use the normalized input vector to find the
# closest node
closest_node = SOM_closest_node(SOM_input_vector);

# Update the winning node and neighborhood values
SOM_update(closest_node, SOM_input_vector);

return 0;
}
```

The neighborhood functions currently implemented include modified Gaussian function and modified Ricker wavelet. Both neighborhood functions are discreet and use Chebyshev distance from the winning neuron and the overall size of the neighborhood to compute an additional multiplier for the learning factor alpha. The neighborhood functions are structured so that they can be easily modified, and additional functions written if needed.

The listing in 4.2 demonstrates a stripped down version of the shell scripts that can be used for instructing the system on what parameters to use when conducting learning. The initial invocation of bro in the shell script creates SOM_saved Bro script file which include a SOM lattice of X,Y dimensions and default features as documented in SOM_config.bro and also commences first ordering pass with alpha and neighborhood

values as included in *alpha* and *neighborhood*. The files *[alpha]* and *neighborhood* contain redefinitions of variables in the Bro format: *"redef alpha = value;"* and *"redef neighborhood = value;"*, respectively.

Listing 4.2. Example shell script used for learning

```
# Initialization and ordering (Shortened for listing)
/home/mmmattik/Bro-Installation/bin/bro -r ~/data.pcap
learn alpha neighborhood sizexX sizeyY;

# Fine tuning phase (Shortened for this listing)
/home/mmmattik/Bro-Installation/bin/bro -r ~/data.pcap
SOM_saved learn alpha002 neighborhood9.bro;
/home/mmmattik/Bro-Installation/bin/bro -r ~/data.pcap
SOM_saved learn alpha002 neighborhood3.bro;
```

4.4 Detection mode

In the detection mode, the system does not modify the SOM lattice based on the input. The SOM lattice, normalization values and used features are provided in the ASCII file created and modified by the system during the learning phase. This file contains everything that has been learned in the learning phase. Portability of the file makes it very easy to change monitoring system if needed, just by copying or moving the file, no databases are used. The information provided in the file is read into memory, and the detection can begin. No file system operations are done by the extension module after this in the detection mode. Detection mode processes are depicted in Figure 6, omitting the initialization and termination processes.

In detection mode, input data is normally processed into a input vector and compared to the existing lattice. If the nearest neuron by the measure Euclidean distance between weight and input vector is more than the threshold value apart, a possible anomaly is reported. In equation 7, the d_{euc} is defined in equation 3 and T_v stands for threshold value that is to be configured by the user when the system is invoked in detection mode.

$$d_{euc}(i,j) > T_v \tag{7}$$

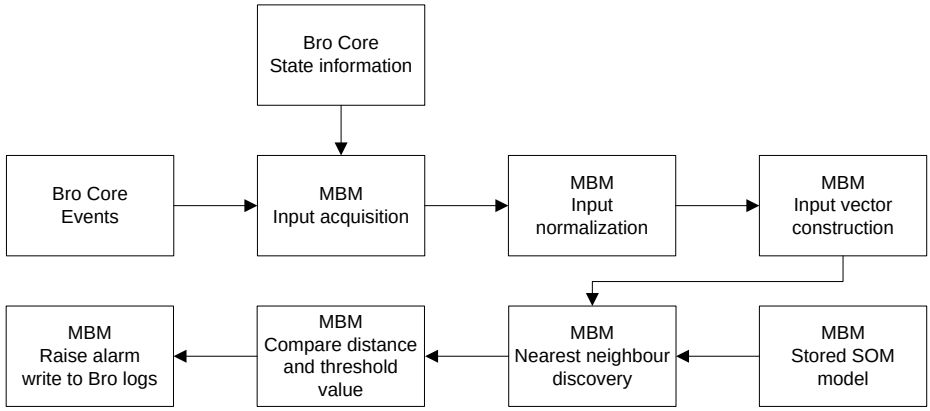

Fig 6. The SOM extension module detection process.

The value of the threshold variable should be as low as possible to limit the amount of f_n, but high enough to limit the number of f_p. The main function of the detection logic is presented in Listing 4.3. The fine tuning of the threshold value has no exact process for it. It needs to be adjusted after the learning phase during the initial detection phase testing.

In addition to the operational detection phase, the detection mode is also needed for testing the results of the learning phase. The amount of false positives and negatives when testing with traces from the target environment provides information on whether the lattice has been properly formed and has converged correctly. Traces not used for the learning phase should be used, along with the traces used for learning. If any of the clean traces produce a significant number of f_p there is possibly something wrong with the convergence of the lattice, the traces or even the initial assumptions concerning the nature of the network environment to be monitored. The threshold value might also be too low for fluctuations in the network traffic of the target environment.

Listing 4.3. Detection logic debugging prints removed and lines wrapped for improved readability

```
##################################################
# Functionality for running the detection
#
function SOM_detect(
c: connection, proto: transport_proto): bool
```

```
{
local SOM_input_vector: vector of double;
local weight_vector: vector of double;
local closest_node: vector of count;
local SOM_distance: double = 0;
local a: count = 0;
local b: count = 0;
local d: count = 0;

# Obtain the input and form normalized input vector
SOM_input_vector = SOM_build_input_vector(c, proto);

# Use input vector to find closest node
closest_node = SOM_closest_node(SOM_input_vector);

for ( index in SOM_counter_depth )
        {
        d=int_to_count(index);
        weight_vector[index] = SOM_model[closest_node[0],
        closest_node[1],d];
        }

SOM_distance = euclidean_distance(
SOM_input_vector, weight_vector);

# Wether threshold was crossed
if ( SOM_distance > delta )
        {
        NOTICE([$note=Possible_Anomaly_Seen,
        $msg=fmt("Distance from nearest node[%d,%d]:[%f]",
        closest_node[0],closest_node[1],SOM_distance),
        $conn=c]);

        return T;
        }
```

```
# Return false for no anomaly
return F;
}
```

The test results for the detection mode using Printocent printed intelligence pilot factory traffic captures both clean and with simulated reconnaissance attacks embedded in some of them are presented in publication VII. The tests did produce false positives. At least part of them were likely due to the nature of the Printocent pilot factory as a relatively volatile research facility. Fine tuning of the system for the particular environment would need to be investigated. This could include such things as selection of other features or even writing new algorithm and event combinations to implement the EMLAD -concept to a fuller degree. However, these actions, further tests and their results remain future work.

Due to the method of operation of the current PoC system, we only catch anomalies after the connection terminates. For the anomalies used in work VII, this was not an issue, as a significant amount of short-lived connections were triggered and recognized as such. The result would likely have been near real-time detection. However, this would not be a case for a connection which is long-lived and includes anomalous traffic. In that case, the anomaly would be detected only at the termination of the connection. This would provide the administrators help in the forensic phase of their work, but confer limited assistance in the detection of the anomaly as it happens. This is to be ameliorated by further development of the system, such as adding the handling of new connection events and more. Currently, only the connection termination event is handled.

4.5 Extendability

The system is structured to allow for easy extendability and to implement new features and functionalities. The attributes governing the function of the abilities already present can be redefined by the user without touching the main scripts. This redefinition is done in the normal Bro fashion, by introducing new script files which *redef* the targeted variables. The core Bro functionality does not need to be touched. The extension and modification of the system only requires action on the extension module specific scripts. Therefore, to use the system, only normal Bro installation of the correct version is needed. As of this writing, that is the version 2.2 released during the latter half of 2013.

The system is also easily extended due to the nature of the Bro NSM. The MBM

system currently only handles the connection termination event which the Bro event engine invokes. Adding functionality to catch additional Bro event engine created events is a straightforward process. In addition to the ML functionality, the normal Bro system remains untouched, allowing the user to combine the MBM with any scripts needed.

However, catching additional events currently requires the user to write the scripts for handling them. In the MBM extension module, there are no additional events available as of January 2014, though several are under development, the idea being that the user could select on which events to handle based on the nature of the underlying network and its behavior.

Implementation of additional features, handling of events and machine learning algorithms is currently under way.

4.6 Future direction

Currently, the module is structured to fashion a single SOM -lattice for all of the IP traffic seen. It only handles the `connection_state_remove` event. Work is on-going to further break this down into separate lattices for separate events, such as transport protocol specific events, and later to application layer protocols. Work is also on-going to allow each different event to be handled by a different machine learning algorithm with a separate feature set, or even several machine learning algorithms with a separate feature set for each. This would allow for more specific features to be developed and increase the efficiency of the system in detecting anomalies through more specific ML models such as SOM lattices. For e.g. `new_connection` events we do not have the same information available as for the connection termination, e.g. connection duration.

New instance of a selected algorithm, such as SOM, would be needed for different feature vectors. Alternatively, whether a single ML instance could be used for different events by looking at only the certain locations in the stored models could also be investigated.

The work presented by Lin *et al.* (2013) describes the DNP3 protocol parser for Bro. New ICS protocol parsers for Bro system are important for the continuation of the system and therefore of great interest. The ability to parse most ICS protocols would enable the development of a machine learning solution which uses protocol specific information as a basis of additional features. An interesting approach would be to implement features or new approaches leveraging the information presented in the paper by Goldenberg & Wool (2013) for Modbus/TCP. Similarly, an approach could be

tailored for DNP3 or other ICS protocols with parsers becoming available for Bro.

Advances in machine learning, such as the deep learning approaches discussed briefly in Section 3.1.4 will possibly provide new approaches to be used in anomaly detection in network security monitoring context.

Improving the extendability by allowing the user to choose between several machine learning schemes or combinations of them would also increase the flexibility. Currently, the system has shortcomings, being a limited PoC implementation. Work is in progress to add functionality to embody the EMLAD concept and generally extend the PoC system.

A complete rewrite of the system is under work to provide a more modular system for handling different Bro core generated events with user configurable ML functionality following the EMLAD concept.

A deep learning Bengio *et al.* (2013), Bengio (2009) functionality is also being actively investigated and could provide very interesting results. However, the deep learning functionality is something that will likely be investigated only after the complete modular structure for selecting different algorithms, events and features is ready. The completion of this machine learning framework module for Bro is foreseen to take considerable time.

5 Discussion and summary

Efficient network security monitoring in any ICS network of any importance is a thing to be taken seriously. This issue is highlighted by the fact that ICS systems power a significant portion of the developed world's critical infrastructures, industrial sites and various other sensitive services. Fortunately, the ICS networks typically exhibit special characteristics such as predictability that can be exploited for improved security monitoring processes. Part of these attributes derive from the restricted and static nature of the ICS networks. These attributes can also be leveraged for NSM in other environments which exhibit them in a much similar fashion.

In this thesis, an industrial control system networks was used as a case example of restricted networks. Additionally, an existing ICS network was used to provide a concrete example and to develop and validate the initial PoC implementation with single event and the SOM algorithm handling.

Further implementation of the approach and testing it in other restricted networks is needed to validate the more generalized approach of the proposed event-driven machine learning concept.

There exist several challenges for performing NSM in the ICS and more generic restricted networks. A number of identified challenges are presented in Section 5.1. The increasing area of exposed attack surface of the ICS and other restricted networks highlights the importance of solving these and other issues that might stand in the way of proper NSM in these environments.

The next steps on using machine learning for anomaly detection and more general network security monitoring is briefly discussed in Section 5.2. Whether the MBM system code base of Bro scripts will be made available publicly depends on several factors which still remain undecided. The system is also undergoing massive changes from a SOM centric PoC extension to a more general ML approach, as explained in Section 4.6 which is specific to the future of the MBM system.

The investigations of this thesis further validate the machine learning based approach for anomaly detection in ICS networks, or other network exhibiting similar characteristics. The results converge with the work by other authors, as demonstrated in Chapter 3. A set of possible features for ML based anomaly detection system was initially investigated by looking at the challenges as presented in Sommer & Paxson

(2010). The investigations were then expanded in papers II and III using traffic captures from an industrial site in Finland. Some of the features were deemed difficult to use and some feasible in the particular environment.

The PoC implementation of the MBM system with the handling for SOM algorithm and a single Bro NSM core produced event was used to demonstrate the viability of using a Bro system and its script language in this manner. The flexibility of the Bro and the richness of its stored network state and introspection functionality make for a good framework to develop the system further towards true operational capability. No similar system to MBM that would have been created using Bro NSM scripts was found during the literature review. The system is also easily used and configured by any user proficient in basic script writing and comfortable with operating a computer system from CLI. For validating the PoC implementation, access was gained to the Printocent facility located in Oulu and several months of traffic was captured for testing purposes.

After the successful validation of the initial PoC, the MBM system is being developed to properly embody the concept of event-driven anomaly detection specifically for ICS networks. The approach has been initially introduced in paper IV, and with more clarity in paper VII. In the form it is presented, the concept appears novel, and will warrant more investigations into the implications of its discovery in the future.

5.1 Challenges

One of the main challenges for more generic work on restricted network security monitoring and anomaly detection remains the issue of how to acquire suitable network traces. By our definitions in Table 1, restricted networks are more controlled than open ones, and therefore traces from them are also more difficult to obtain. How to obtain suitable traces from the industry vary of their intellectual property and defaulting to the answer of "No" is an issue to be solved in the future. This requires tangible proof of potential benefits from the approach of NSM and anomaly detection in particular. This thesis represents research work aimed at improving the security of the operators of these restricted networks. It is hoped that it can be used to further convince the administrators and managers who are in charge to provide access to relevant network traces for their potential collaborators and partners.

An issue causing difficulties to anomaly detection in general is the increasing level of encryption used in a variety of networks (Bejtlich 2004). Today, industrial control systems protocols typically lack encryption, e.g. Modbus, DNP3 and ICCP (Knapp

2011). This enables the use of anomaly detection approaches requiring access to the plain text payload. The advancement of ubiquitous encryption into the ICS domain will likely be slow, due to the latencies introduced by techniques such as block encryption (Weiss 2010).

In the more generic setting, the increase in the amount of data load of a given network is causing anomaly detection difficulties (Bejtlich 2004). This can also become an issue for NSM done in the context ICS networks. This depends on the increase of connectivity, remote connections and protocols used. In general, it is not a major concern in the networks such as the Printocent control network investigated in paper IV and paper VII belonging to this thesis. However, increase in traffic loads creates issues for computationally expensive algorithms such as SOM with a large lattice and number of features. When deciding on the algorithm and design used for ML anomaly detection, the platform and data loads need to be taken carefully into account. The system needs to be able to perform online, not just on captured traces.

The approach PoC with the initial implementation of MBM system works well for the task it is meant for. However, there are shortcomings. The approach of judging connection based on the information after they have terminated creates a situation, where the events might go unnoticed for a good period of time. However, it must be mentioned that the injected network anomalies that the system was tested with in paper VII all created a sudden surge of short-lived terminating connections and were thus detected rapidly. Bro currently only supports single core even on a multi-core computer. This creates a requirement for a cluster deployment even if the machine would have performance reserves in the sense of idling cores and free memory. A deployment of a scalable NIDS with cluster architecture is introduced in Vallentin *et al.* (2007). The paper focuses on the cluster deployment using commodity hardware. Cluster deployment of Bro NSM is supported in the version 2.2.

Other challenges exist as well, and new ones are constantly evolving. The evolving nature of cybersecurity is an important characteristic of the field. Cybersecurity is a process, not a static state or a product (Schneier 2009).

5.2 Conclusion

The security requirements of the ICS networks and by the extension the ICS and CI themselves is not likely to diminish in the near future. The progress of increasing connectivity combined with the persistent cyber fragility of the ICS is also likely

to create new challenges. Even without possible new challenges, the existing ones continue to pose problems. Solving these existing issues includes the development and deployment of proper security monitoring mechanisms and one such mechanism is anomaly detection. Network security monitoring and ICS networks were discussed in Chapter 2. (Knapp 2011, Linda *et al.* 2012, Langner 2011a, Bruner 2013, Hadeli *et al.* 2009, Bejtlich 2013)

As anomaly detection is an important aspect of the overall network security monitoring, and especially suited for ICS networks as discussed in Section 3.4, advances in the field can be translated into real security gains. The progress of anomaly detection in ICS networks will also likely benefit from the advances made in other fields, e.g. big data analytics Bruner (2013). The new algorithms and approaches investigated can possibly be leveraged for securing the ICS and therefore CI networks. This can also include possibilities offered by approaches such as deep learning, explained in Section 3.1.4. Anomaly detection using machine learning methods was discussed in Chapter 3.

As mentioned in Section 4.6, the MBM module PoC for Bro is being rewritten to enable use of different machine learning algorithms in addition to SOM. The aim is to extend the system to a user configurable general machine learning framework for Bro in which the user can define the learning algorithms used and Bro core events handled and the approach. The MBM system will also be geared towards a more generalized restricted network approach. This requires considerable advances in obtaining access to suitable network environments. Environments such as Printocent are comparatively easy to gain access to, being controlled by a research institute. The initial MBM implementation is presented in Chapter 4.

The investigations of this thesis, based on the publications I-VII contribute to the advancement of anomaly detection and network security monitoring in ICS environments. This is accomplished through theoretical and literature studies, as well as a concrete PoC system for testing and further development. The way ICS and CI are increasingly exposed to the outside world will continue to drive the need for further research in this area.

References

Adler A, Mayhew M, Cleveland J, Atighetchi M & Greenstadt R (2013) Using machine learning for behavior-based access control: Scalable anomaly detection on tcp connections and http requests. Proc. Military Communications Conference, MILCOM 2013 - 2013 IEEE, 1880–1887.

Alcaraz C, Roman R, Najera P & Lopez J (2013) Security of industrial sensor network-based remote substations in the context of the internet of things. Ad Hoc Networks 11(3): 1091 – 1104.

Anderson RJ (2008) Security Engineering: A Guide to Building Dependable Distributed Systems. Wiley Publishing, 2 edition.

Barto A, Sutton R & Anderson C (1983) Neuronlike adaptive elements that can solve difficult learning control problems. Systems, Man and Cybernetics, IEEE Transactions on SMC-13(5): 834–846.

Basecamp, (accessed 26/3/2014) http://www.digitalbond.com/tools/basecamp/.

Bejtlich R (2004) The Tao Of Network Security Monitoring: Beyond Intrusion Detection. Addison-Wesley Professional.

Bejtlich R (2013) The Practice of Network Security Monitoring. No Starch Press.

Bengio Y (2009) Learning deep architectures for AI. Foundations and Trends in Machine Learning 2(1): 1–127. Also published as a book. Now Publishers, 2009.

Bengio Y, Courville A & Vincent P (2013) Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8): 1798–1828.

Beran J, Fiedler P & Zezulka F (2010) Virtual automation networks. Industrial Electronics Magazine, IEEE 4(3): 20–27.

Bro NSM, (accessed 2/12/2013) http://www.bro.org/.

Bruner J (2013) Industrial Internet. O'Reilly Media.

Callegari C, Coluccia A, D'Alconzo A, Ellens W, Giordano S, Mandjes M, Pagano M, Pepe T, Ricciato F & Zuraniewski P (2013) A methodological overview on anomaly detection. In: Biersack E, Callegari C & Matijasevic M (eds) Data Traffic Monitoring and Analysis, volume 7754 of *Lecture Notes in Computer Science*, 148–183. Springer Berlin Heidelberg.

Chandola V, Banerjee A & Kumar V (2009) Anomaly detection: A survey. ACM Comput. Surv. 41(3): 15:1–15:58.

Debar H, Becker M & Siboni D (1992) A neural network component for an intrusion detection system. Proc. Research in Security and Privacy, 1992. Proceedings., 1992 IEEE Computer Society Symposium on, 240–250.

Denning D (1987) An intrusion-detection model. Software Engineering, IEEE Transactions on SE-13(2): 222 – 232.

Dreger H, Feldmann A, Paxson V & Sommer R (2008) Predicting the resource consumption of network intrusion detection systems. In: Lippmann R, Kirda E & Trachtenberg A (eds) Recent Advances in Intrusion Detection, volume 5230 of *Lecture Notes in Computer Science*, 135–154. Springer Berlin / Heidelberg.

Dreger H, Kreibich C, Paxson V & Sommer R (2005) Enhancing the accuracy of network-based intrusion detection with host-based context. In: Julisch K & Kruegel C (eds) Detection of Intrusions and Malware, and Vulnerability Assessment, volume 3548 of *Lecture Notes in*

Computer Science, 584–600. Springer Berlin / Heidelberg. 10.1007/11506881_13.

Dzung D, Naedele M, von Hoff T & Crevatin M (2005) Security for industrial communication systems. Proceedings of the IEEE 93(6): 1152–1177.

ELSA, (accessed 1/4/2014) https://code.google.com/p/enterprise-log-search-and-archive/.

Ericsson G (2010) Cyber security and power system communication x2014;essential parts of a smart grid infrastructure. Power Delivery, IEEE Transactions on 25(3): 1501–1507.

Erwin E, Obermayer K & Schulten K (1992) Self-organizing maps: Ordering, convergence properties and energy functions. Biological Cybernetics 67: 47–55.

Fiore U, Palmieri F, Castiglione A & Santis AD (2013) Network anomaly detection with the restricted boltzmann machine. Neurocomputing 122(0): 13 – 23. Advances in cognitive and ubiquitous computing.

García-Teodoro P, Díaz-Verdejo J, Maciá-Fernández G & Vázquez E (2009) Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security 28(1-2): 18 – 28.

Goldenberg N & Wool A (2013) Accurate modeling of modbus/tcp for intrusion detection in SCADA systems. International Journal of Critical Infrastructure Protection 6(2): 63 – 75.

Gonzalez J & Paxson V (2006) Enhancing network intrusion detection with integrated sampling and filtering. In: Zamboni D & Kruegel C (eds) Recent Advances in Intrusion Detection, volume 4219 of *Lecture Notes in Computer Science*, 272–289. Springer Berlin / Heidelberg. 10.1007/11856214_14.

Hadeli H, Schierholz R, Braendle M & Tuduce C (2009) Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. Proc. Emerging Technologies Factory Automation, 2009. ETFA 2009. IEEE Conference on, 1–8.

Haykin S (2009) Neural Networks and Learning Machines. Pearson International Edition. Pearson Education, Limited.

Hirasawa K, Okubo M, Katagiri H, Hu J & Murata J (2001) Comparison between genetic network programming (gnp) and genetic programming (gp). Proc. Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, 2: 1276–1282 vol. 2.

Honda S, Nakashima T & Oshima S (2008) Entropy based analysis of anomaly access of ip packets. Proc. Innovative Computing Information and Control, 2008. ICICIC '08. 3rd International Conference on, 101–101.

Hu W, Xie D, Tan T & Maybank S (2004) Learning activity patterns using fuzzy self-organizing neural network. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 34(3): 1618–1626.

Igure VM, Laughter SA & Williams RD (2006) Security issues in {SCADA} networks. Computers & Security 25(7): 498 – 506.

Issariyapat C & Fukuda K (2009) Anomaly detection in ip networks with principal component analysis. Proc. Communications and Information Technology, 2009. ISCIT 2009. 9th International Symposium on, 1229–1234.

Kayacik H, Zincir-Heywood A & Heywood M (2003) On the capability of an som based intrusion detection system. Proc. Neural Networks, 2003. Proceedings of the International Joint Conference on, 3: 1808–1813 vol.3.

Kayacik H, Zincir-Heywood A & Heywood M (2007) A hierarchical som-based intrusion detection system. Eng. Appl. Artif. Intell. 20(4): 439–451.

Knapp E (2011) Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems. Elsevier Science.

Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biological Cybernetics 43(1): 59–69.

Kohonen T, Hynninen J, Kangas J & Laaksonen J (1996) SOM PAK: The Self-Organizing Map program package.

Kohonen T, Schroeder MR & Huang TS (eds) (2001) Self-Organizing Maps. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edition.

Kortuem G, Kawsar F, Fitton D & Sundramoorthy V (2010) Smart objects as building blocks for the internet of things. Internet Computing, IEEE 14(1): 44–51.

Langner R (2011a) Robust Control System Networks: How to Achieve Reliable Control After Stuxnet. Momentum Press.

Langner R (2011b) Stuxnet: Dissecting a cyberwarfare weapon. Security Privacy, IEEE 9(3): 49–51.

Lee S & Heinbuch D (2001) Training a neural-network based intrusion detector to recognize novel attacks. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 31(4): 294–299.

Lin H, Slagell A, Di Martino C, Kalbarczyk Z & Iyer RK (2013) Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol. Proc. Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, ACM, New York, NY, USA, 5:1–5:4.

Linda O, Manic M & Vollmer T (2012) Improving cyber-security of smart grid systems via anomaly detection and linguistic domain knowledge. Proc. Resilient Control Systems (ISRCS), 2012 5th International Symposium on, 48–54.

Linda O, Manic M, Vollmer T & Wright J (2011) Fuzzy logic based anomaly detection for embedded network security cyber sensor. Proc. Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on, 202–209.

Linda O, Vollmer T & Manic M (2009) Neural network based intrusion detection system for critical infrastructures. Proc. Proceedings of the 2009 international joint conference on Neural Networks, IEEE Press, Piscataway, NJ, USA, 102–109.

Liu Y, Ning P & Reiter MK (2009) False data injection attacks against state estimation in electric power grids. Proc. Proceedings of the 16th ACM Conference on Computer and Communications Security, ACM, New York, NY, USA, 21–32.

Mabu S, Chen C, Lu N, Shimada K & Hirasawa K (2011) An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41(1): 130–139.

Manocha S & Girolami M (2007) An empirical analysis of the probabilistic k-nearest neighbour classifier. Pattern Recognition Letters 28(13): 1818 – 1824.

Mattern F & Floerkemeier C (2010) From the internet of computers to the internet of things. In: Sachs K, Petrov I & Guerrero P (eds) From Active Data Management to Event-Based Systems and More, volume 6462 of *Lecture Notes in Computer Science*, 242–259. Springer Berlin Heidelberg.

Metasploit, (accessed 26/3/2014) http://www.metasploit.com/.

Morris T, Vaughn R & Dandass Y (2012) A retrofit network intrusion detection system for modbus rtu and ascii industrial control systems. Proc. System Science (HICSS), 2012 45th Hawaii International Conference on, 2338–2345.

Morris TH, Jones BA, Vaughn RB & Dandass YS (2013) Deterministic intrusion detection rules for modbus protocols. Proc. System Sciences (HICSS), 2013 46th Hawaii International

Conference on, 1773–1781.
Moslehi K & Kumar R (2010) A reliability perspective of the smart grid. Smart Grid, IEEE Transactions on 1(1): 57–64.
Mukherjee B, Heberlein L & Levitt K (1994) Network intrusion detection. Network, IEEE 8(3): 26–41.
Mukkamala S, Janoski G & Sung A (2002) Intrusion detection using neural networks and support vector machines. Proc. Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on, 2: 1702–1707.
Muniyandi AP, Rajeswari R & Rajaram R (2012) Network anomaly detection by cascading k-means clustering and c4.5 decision tree algorithm. Procedia Engineering 30(0): 174 – 182. International Conference on Communication Technology and System Design 2011.
Murphy KP (2012) Machine Learning: A Probabilistic Perspective. The MIT Press.
Nessus Vulnerability Scanner ((accessed 1/2/2014)) http://www.tenable.com/products/nessus/.
Nicholson A, Webber S, Dyer S, Patel T & Janicke H (2012) SCADA security in the light of cyber-warfare. Computers & Security 31(4): 418 – 436.
Nikto2 Web Server Scanner ((accessed 2/3/2014)) https://www.cirt.net/nikto2/.
Nmap Network Security Scanner ((accessed 7/2/2013)) http://www.nmap.org/.
Okhravi H & Nicol DM (2009) Application of trusted network technology to industrial control networks. International Journal of Critical Infrastructure Protection 2(3): 84 – 94.
Om H & Kundu A (2012) A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. Proc. Recent Advances in Information Technology (RAIT), 2012 1st International Conference on, 131–136.
Parikh D & Chen T (2008) Data fusion and cost minimization for intrusion detection. Information Forensics and Security, IEEE Transactions on 3(3): 381–389.
Paxson V (1999) Bro: a system for detecting network intruders in real-time. Computer Networks 31(23-24): 2435 – 2463.
Pollet J (July 2010) Electricity for free? The dirty underbelly of SCADA and smart meters. Proc. 2010 BlackHat Technical Conference.
PrintoCent, (accessed 1/6/2013) http://www.printocent.net.
Prokhorov D & Wunsch D (1997) Adaptive critic designs. Neural Networks, IEEE Transactions on 8(5): 997–1007.
Ramadas M, Ostermann S & Tjaden B (2003) Detecting anomalous network traffic with self-organizing maps. Proc. In Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection, LNCS, Springer Verlag, 36–54.
Rexworthy B (2009) Intrusion detections systems âĂŞ an outmoded network protection model. Network Security 2009(6): 17 – 19.
Russell SJ & Norvig P (2010) Artificial Intelligence: A Modern Approach. Prentice Hall, 3rd edition.
Sarasamma S, Zhu Q & Huff J (2005) Hierarchical kohonenen net for anomaly detection in network security. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 35(2): 302–312.
Schneier B (2009) Schneier on Security. Wiley.
Security Onion, (accessed 14/5/2014) http://www.securityonion.net.
Shodan, (accessed 26/3/2014) http://www.shodanhq.com/.
Shon T, Kim Y, Lee C & Moon J (2005) A machine learning framework for network anomaly detection using svm and ga. Proc. Information Assurance Workshop, 2005. IAW '05.

Proceedings from the Sixth Annual IEEE SMC, 176–183.
Shon T & Moon J (2007) A hybrid machine learning approach to network anomaly detection. Information Sciences 177(18): 3799 – 3821.
Sindhu SSS, Geetha S & Kannan A (2012) Decision tree based light weight intrusion detection using a wrapper approach. Expert Systems with Applications 39(1): 129 – 141.
Snort, (accessed 1/12/2013) http://www.snort.org/.
Sommer R & Paxson V (2010) Outside the closed world: On using machine learning for network intrusion detection. Proc. Security and Privacy (SP), 2010 IEEE Symposium on, 305 –316.
Suricata, (accessed 1/12/2013) http://suricata-ids.org/.
Sutic D & Atlagic B (2013) Requirement bottlenecks in a cloud based scada system. Proc. Information Communication Technology Electronics Microelectronics (MIPRO), 2013 36th International Convention on, 857–862.
Tcpdump, (accessed 7/6/2013) http://www.tcpdump.org/.
Thottan M & Ji C (2003) Anomaly detection in ip networks. Signal Processing, IEEE Transactions on 51(8): 2191–2204.
Tsai CF, Hsu YF, Lin CY & Lin WY (2009) Intrusion detection by machine learning: A review. Expert Systems with Applications 36(10): 11994 – 12000.
Tsai CF & Lin CY (2010) A triangle area based nearest neighbors approach to intrusion detection. Pattern Recognition 43(1): 222 – 229.
Tylman W (2008) Anomaly-based intrusion detection using bayesian networks. Proc. Dependability of Computer Systems, 2008. DepCos-RELCOMEX '08. Third International Conference on, 211–218.
Valdes A & Cheung S (2009) Intrusion monitoring in process control systems. Proc. System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on, 1–7.
Vallentin M, Sommer R, Lee J, Leres C, Paxson V & Tierney B (2007) The nids cluster: Scalable, stateful network intrusion detection on commodity hardware. In: Kruegel C, Lippmann R & Clark A (eds) Recent Advances in Intrusion Detection, volume 4637 of *Lecture Notes in Computer Science*, 107–126. Springer Berlin / Heidelberg. 10.1007/978-3-540-74320-0_6.
Vapnik V (1999) An overview of statistical learning theory. Neural Networks, IEEE Transactions on 10(5): 988–999.
Wagner A & Plattner B (2005) Entropy based worm and anomaly detection in fast ip networks. Proc. Enabling Technologies: Infrastructure for Collaborative Enterprise, 2005. 14th IEEE International Workshops on, 172–177.
Weaver N, Paxson V & Sommer R (2006) Work in progress: Bro-lan pervasive network inspection and control for lan traffic. Proc. Securecomm and Workshops, 2006, 1 –2.
Weiss J (2010) Protecting Industrial Control Systems from Electronic Threats. Momentum Press.
Williams TJ (1994) The purdue enterprise reference architecture. Computers in Industry 24(2 - 3): 141 – 158.
Willshaw DJ & Von Der Malsburg C (1976) How Patterned Neural Connections Can Be Set Up by Self-Organization. Proceedings of the Royal Society of London. Series B, Biological Sciences 194(1117): 431–445.
Wireshark, (accessed 2/5/2013) http://http://www.wireshark.org/.
Yan Y, Qian Y, Sharif H & Tipper D (2012) A survey on cyber security for smart grid communications. Communications Surveys Tutorials, IEEE 14(4): 998–1010.
Yang D, Usynin A & Hines J (2006) Anomaly-based intrusion detection for scada systems. Proc. Proceedings of the 5th Intl. Topical Meeting on Nuclear Plant Instrumentation, Control and

Human Machine Interface Technologies.

Ye N, Zhang Y & Borror C (2004) Robustness of the markov-chain model for cyber-attack detection. Reliability, IEEE Transactions on 53(1): 116–123.

Zhang J & Moore A (2007) Traffic trace artifacts due to monitoring via port mirroring. Proc. End-to-End Monitoring Techniques and Services, 2007. E2EMON '07. Workshop on, 1–8.

Original publications

I Mantere M & Uusitalo I & Sailio M & Noponen S (2012) Challenges of Machine Learning Based Monitoring for Industrial Control System Networks. Proceedings of 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA 2012).

II Mantere M & Sailio M & Noponen S (2012) Feature Selection for Machine Learning Based Anomaly Detection in Industrial Control System Networks. Proceedings of 2012 IEEE International Conference on Green Computing and Communications (GreenCom 2012): 771-774. Presented in 2nd workshop on Security of Systems and Software resiLiency (3SL 2012) organized during GreenCom.

III Mantere M & Sailio M & Noponen S & (2013) Network Traffic Features for Anomaly Detection in Specific Industrial Control System Network. (Future Internet 5(4)): 460-473.

IV Mantere M & Sailio M & Noponen S (2014) A Module for Anomaly Detection in ICS Networks. Proceedings of 3rd ACM International Conference on High Confidence Networked Systems (HiCoNS 2014).

V Mantere M & Noponen S & Olli P & Salonen J (2014) Network Security Monitoring in Small-scale Smart-grid laboratory. Published in the 2nd International Workshop on Emerging Cyberthreats and Countermeasures (ECTCM 2014).

VI Sailio M & Mantere M & Noponen S (2014) Network Security Analysis Using Behavior History Graph. Published in the Industrial Track Workshop at the 9th International Conference on Availability, Reliability and Security (ARES-IND 2014).

VII Mantere M & Sailio M & Noponen S (2014) Detecting Anomalies in Printed Intelligence Factory Network. Published in the 9th International Conference on Risks and Security of Internet and Systems (CRiSIS 2014). In press.

Book orders:
Granum: Virtual book store
http://granum.uta.fi/granum/

UNIVERSITY OF OULU P.O. Box 8000 FI-90014 UNIVERSITY OF OULU FINLAND

ACTA UNIVERSITATIS OULUENSIS

UNIVERSITY of OULU
OULUN YLIOPISTO

ISBN 978-952-62-0814-5 (Paperback)
ISBN 978-952-62-0815-2 (PDF)
ISSN 0355-3213 (Print)
ISSN 1796-2226 (Online)