Tuomo Sipola
Janne Alatalo
Monika Wolfmayr
Tero Kokkonen *Editors*

Artificial Intelligence for Security

Enhancing Protection in a Changing World

Springer

Artificial Intelligence for Security

Tuomo Sipola • Janne Alatalo •
Monika Wolfmayr • Tero Kokkonen
Editors

Artificial Intelligence for Security

Enhancing Protection in a Changing World

Springer

Editors
Tuomo Sipola
Jamk University of Applied Sciences
Jyväskylä, Finland

Janne Alatalo
Jamk University of Applied Sciences
Jyväskylä, Finland

Monika Wolfmayr
Jamk University of Applied Sciences
Jyväskylä, Finland

Tero Kokkonen
Jamk University of Applied Sciences
Jyväskylä, Finland

ISBN 978-3-031-57451-1 ISBN 978-3-031-57452-8 (eBook)
https://doi.org/10.1007/978-3-031-57452-8

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

Artificial intelligence (AI) has seen unprecedented revival in the public eye during the last years. Recent advances, such as AI-based image generation and large language models (e.g., ChatGPT), have demonstrated the potential of this technology. The use of AI technologies to protect the world we live in is topical as new threats emerge. After our previous book, *Artificial Intelligence and Cybersecurity: Theory and Applications* (Springer 2023), we were left with the feeling that not everything had been said about the topic. Consequently, we gathered a group of international experts who could write about the role of AI in the changing world.

This book is divided into three parts, concerning methodological fundamentals, critical infrastructure protection, and anomaly detection. This division emerged from the contents of the chapters that we received. It was quite natural to have chapters about protection of critical infrastructure and about anomaly detection methods for digitalized solutions.

The first part is about methodological fundamentals of artificial intelligence, especially within the scope of security. Adrowitzer et al. provide a blueprint towards a safe world with AI and its development and application. Holmström et al. discuss the use of AI from the point of view of organizational and managerial cybersecurity while evaluating its silver bullet status in the hype discourse. After these general introductions to the topic, we turn to deeper investigations about AI methodology. Data are the raw material of most AI work, and protecting the privacy of the individuals whose data are used is an important concern. Kilpala and Kärkkäinen present a review of ways to evaluate differential privacy models. Furthermore, Van Gerwen et al. discuss the challenges of explainable AI in the context of threat intelligence. Jansevskis and Osis, on the other hand, explain knowledge discovery frameworks and how to include security considerations into them. This relates back to the importance of data and knowledge extraction. This part is concluded by the chapter written by Glazunov and Zarras, which considers the robustness of deep learning. This chapter focuses on technical details, presenting multiple attacks and their significance.

The second part is about the use of artificial intelligence for critical infrastructure protection. As an introduction to the topic, Nweke and Yayilgan explore the use

of AI for the protection of cyber-physical systems. After this, we continue with domain-specific studies. As a first example, Rasmus studies the use of AI tools for small enterprises in the context of security. Another domain is covered by Kiviharju in the chapter about cybersecurity for logistics. Energy, communications, and healthcare are perhaps the most well-known examples of critical infrastructure. Consequently, Martinelli et al. continue with a study about protecting smart grids. Zolotukhin et al. discuss the protection of mobile networks against adversarial examples. Finally, Jonske et al. introduce the reader to the healthcare domain in their chapter about teaching machine learning with medical data.

The last part of the book includes three chapters about artificial intelligence for anomaly detection in various scenarios. Falzone et al. emphasize the importance of automated monitoring of log data and demonstrate the use of AI to detect anomalies in real time. Shahrivar and Millar detect attacks in large-scale event data using machine learning. The book is concluded by Alqarni and Azim who use deep learning to detect anomalies in Internet of Things (IoT) networks.

This book is useful to professionals who are interested in using artificial intelligence for security purposes. It will also be helpful to those who have concerns about its use in the various industry domains. Understanding latest advancements in this field should be useful to those who want to understand modern cybersecurity in detail, and especially to experts in the field, who want to follow research and the latest trends.

Two conflicts of interest should be disclosed. First, Kai Rasmus is supervised in his PhD studies by one of the editors, Tero Kokkonen. Second, Mansour Alqarni's place of affiliation, Fanshawe College, has commercial co-operation related to the cyber range at the editors' institution. These chapters have undergone the same editorial process as all the other chapters in this book.

We would like to thank the authors for sacrificing their time to provide our book with interesting and topical chapters. Without their willingness and valuable contributions, this book would not have become a reality. We wish to extend our acknowledgments to the reviewers who have ensured the relevance and quality of the chapters. The review committee is listed in the front matter, except for the reviewers who wished to remain anonymous.

New, rapidly developing technologies present us with new challenges. Artificial intelligence does not differ in this regard, and the security aspects of the changes it brings should be noticed, so that we can build more secure systems. We hope this book provides the reader unique perspectives to enhancing protection in these circumstances.

Jyväskylä, Finland
November, 2023

Tuomo Sipola
Janne Alatalo
Monika Wolfmayr
Tero Kokkonen

Contents

Contributors

Alexander Adrowitzer Department Computer Science and Security, St. Pölten University of Applied Sciences, St. Pölten, Austria

Daniel Innala Ahlmark Luleå University of Technology, Luleå, Sweden

Julian Alff Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Mansour Alqarni Fanshawe College, London, ON, Canada
Ontario Tech University, Oshawa, ON, Canada

Simon Andersson Luleå University of Technology, Luleå, Sweden

Akramul Azim Ontario Tech University, Oshawa, ON, Canada

Fin H. Bahnsen Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Oleh Bakumenko Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Marcel Birnbach Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Alexander Buchelt Department Computer Science and Security, St. Pölten University of Applied Sciences, St. Pölten, Austria

Jorge Constantino Delft University of Technology, Delft, Netherlands

Amin Dada Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Jan Egger Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Oliver Eigner Department Computer Science and Security, St. Pölten University of Applied Sciences, St. Pölten, Austria

Åsa Ericson Luleå University of Technology, Luleå, Sweden

Simone Falzone AEB SE, Stuttgart, Germany

Jana Fragemann Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Sarah van Gerwen Vrije Universiteit, Amsterdam, Netherlands

Misha Glazunov Delft University of Technology, Delft, Netherlands

Gabriele Gühring Esslingen University of Applied Sciences, Esslingen, Germany

Timo Hämäläinen Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

Anton Holmström Luleå University of Technology, Luleå, Sweden

Martins Jansevskis Vidzeme University of Applied Sciences, Valmiera, Latvia

Frederic Jonske Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Benjamin Jung AEB SE, Stuttgart, Germany

Tommi Kärkkäinen Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

Peter Kieseberg Department Computer Science and Security, St. Pölten University of Applied Sciences, St. Pölten, Austria

Minna Kilpala Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

Moon-Sung Kim Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Mikko Kiviharju Computer Science Department, Aalto University, Espoo, Finland

Jens Kleesiek Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Bram Klievink Leiden University the Hague, Den Haag, Netherlands

Maxim Kondratenko Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Johannes Kraus Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Johan Lugnet Luleå University of Technology, Luleå, Sweden

Maria Lymbery Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Fabio Martinelli Institute for Informatics and Telematics, National Research Council of Italy (CNR), Pisa, Italy

Fabio Massacci Vrije Universiteit, Amsterdam, Netherlands
University of Trento, Trento, Italy

Francesco Mercaldo Institute for Informatics and Telematics, National Research Council of Italy (CNR), Pisa, Italy
University of Molise, Campobasso, Italy

Stuart Millar Rapid7 LLC, Boston, MA, USA
Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany

Livinus Obiora Nweke Norwegian University of Science and Technology (NTNU), Trondheim, Norway
Noroff Accelerate, Oslo, Norway

Kaspars Osis Vidzeme University of Applied Sciences, Valmiera, Latvia

Kevin Osthues Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Gregor Pavlin Thales Research and Technology, Delft, Netherlands

Kai Rasmus Luode Consulting Oy, Jyväskylä, Finland

Lars Reinike Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Ritten Roothaert Vrije Universiteit, Amsterdam, Netherlands

Antonella Santone University of Molise, Campobasso, Italy

Stefan Schlobach Vrije Universiteit, Amsterdam, Netherlands

Benjamin Schulz Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Pojan Shahrivar Rapid7 LLC, Boston, MA, USA

Fabian Siethoff Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Tobias Simon Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Marlies Temper Department Computer Science and Security, St. Pölten University of Applied Sciences, St. Pölten, Austria

Katja Tuma Vrije Universiteit, Amsterdam, Netherlands

Ben Wagner Delft University of Technology, Delft, Netherlands

Joey Wang Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Brecht Weerheijm Leiden University the Hague, Den Haag, Netherlands

Sule Yildirim Yayilgan Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Di Zhang Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

Nils Zhang Faculty of Mathematics, University Duisburg-Essen, Essen, Germany

Apostolis Zarras University of Piraeus, Piraeus, Greece

Mikhail Zolotukhin Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland

List of Reviewers

António Abreu Instituto Politécnico de Setúbal, Setúbal, Portugal

Tarja Ajo Jamk University of Applied Sciences, Jyväskylä, Finland

Raquel Barreira Instituto Politécnico de Setúbal, Setúbal, Portugal

Svetlana Boudko Norwegian Computing Center, Oslo, Norway

Panagiotis Douris Center for Security Studies (KEMEA), Athens, Greece

Tapio Frantti University of Jyväskylä, Jyväskylä, Finland

David Hästbacka Tampere University, Tampere, Finland

Jari Hautamäki Jamk University of Applied Sciences, Jyväskylä, Finland

Eppu Heilimo Jamk University of Applied Sciences, Jyväskylä, Finland

Pasi Hyytiäinen Jamk University of Applied Sciences, Jyväskylä, Finland

Petri Kannisto Tampere University, Tampere, Finland

Esther Kern Brandenburg Institute for Society and Security, Potsdam, Germany

Lukas Daniel Klausner St. Pölten University of Applied Sciences, St. Pölten, Austria

Gabriela Labres Mallmann University of Jyväskylä, Jyväskylä, Finland

Antti-Jussi Lakanen University of Jyväskylä, Jyväskylä, Finland

Miguel López Instituto Politécnico de Setúbal, Setúbal, Portugal

Antti Mäkelä Jamk University of Applied Sciences, Jyväskylä, Finland

Ilkka Pölönen University of Jyväskylä, Jyväskylä, Finland

Jouni Pöyhönen University of Jyväskylä, Jyväskylä, Finland

Fabi Prezja University of Jyväskylä, Jyväskylä, Finland

Markus Wurzenberger AIT Austrian Institute of Technology, Vienna, Austria

Part I
Methodological Fundamentals of Artificial Intelligence

Safeguarding the Future of Artificial Intelligence: An AI Blueprint

Alexander Adrowitzer, Marlies Temper, Alexander Buchelt, Peter Kieseberg, and Oliver Eigner

1 Introduction

The history of artificial intelligence begins in the 1950s, the first time the term was officially used was in the proposal for the so-called Dartmouth Summer Research Project on Artificial Intelligence in 1956, which was requested by important scientists of the time such as Claude Shannon (the founder of modern information theory), Marvin Minsky, Nathaniel Rochester, and John McCarthy [32]. At that time, the application already contained questions such as "How can a computer be programmed to use a language," something that has now only become possible with language models such as ChatGPT. Even then, people were concerned about the ethical implications of such technology [33, 42, 44, 52], even though it would be decades before the first working applications of artificial intelligence were developed.

Artificial intelligence has found its way into many products of everyday life. Positive examples include vacuum cleaning robots, personal assistants, or assistance systems in cars. Several studies [3, 40] show that such systems bear various concerns, and therefore legal, ethical, and domain-specific considerations have to be made. From this emerges a strong need to safeguard these AI systems from malicious attacks.

When embarking on the development of AI systems, there are established standard processes that can be adopted.

The root for currently used life cycles can be found in data mining processes. A well-known approach is the Knowledge Discovery in Databases (KDD) process [19], which is composed of selecting target data, that is to be analyzed,

A. Adrowitzer (✉) · M. Temper · A. Buchelt · P. Kieseberg · O. Eigner
Department Computer Science and Security, St. Pölten University of Applied Sciences, St. Pölten, Austria
e-mail: alexander.adrowitzer@fhstp.ac.at

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_1

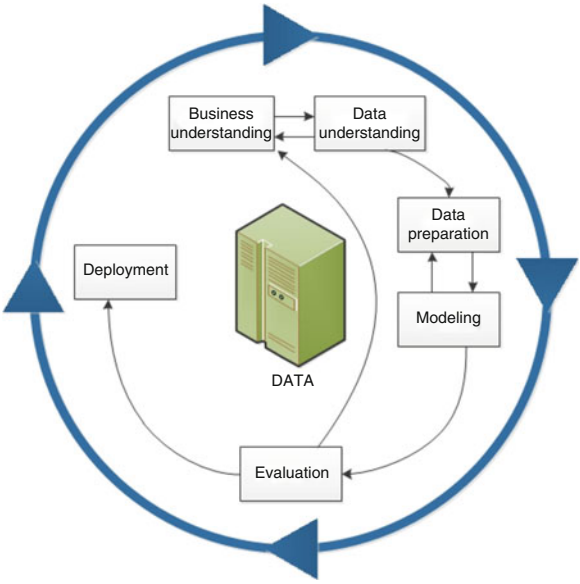


Fig. 1 The six phases of the CRISP-DM lifecycle. Source: https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview

preprocessing the data and performing necessary cleaning steps, transforming the data, mining existing patterns within the data, and finally interpreting what was found. Patterns like this are important as they provide a standard order of operations for practitioners of the field to adhere to. In using them, certain standards like ethical or moral can be enforced.

The most notable successor to the KDD is the CRoss Industry Standard Process for Data mining (CRISP-DM) [45]. It was developed in 1996 and became a European Union project in 1997 with the leadership of five companies. The methodology was presented in 1999 and published as a data mining guide. In subsequent years, discussions were held for updating the model. CRISP-DM is important due to its widespread adoption and several advantages it offers to the data mining industry. It is the most widely used data mining model, providing a structured and systematic approach to data analysis. It helps address existing challenges in data mining by offering a step-by-step guide for practitioners. Figure 1 shows the original CRISP-DM workflow with its major parts. Its industry, tool, and application neutrality contribute to its success, making it adaptable and applicable across different domains. These features make the model a good basis for developing secure AI applications. We will have a closer look at the different phases of the model now: business understanding, data understanding, data preparation, modeling, evaluation, and deployment.

During the business understanding phase, the primary focus is on formulating relevant questions. A skilled data scientist possesses deep knowledge of the application domain they are working in. This domain expertise enables them to identify and articulate questions that can be addressed using analytical methods. While one might assume that finding questions is straightforward, often domain

experts are unaware of existing methods to solve their problems, or they simply go through daily routines without questioning them. When a problem is identified, its initial formulation may prove insufficient and necessitate adjustments.

The data understanding phase involves thoroughly examining the available raw data within an organization. It is crucial to understand both the strengths and limitations of the data in order to effectively work with it. Frequently, data is collected without a specific goal in mind, resulting in potential gaps for addressing certain questions. In such cases, it may be necessary to acquire data from external sources to supplement the existing dataset. If fortunate, the required data might be available as open data; however, there could be associated costs if data needs to be purchased or is not readily accessible. Therefore, the availability of data can directly influence the framing of the question. Data preparation is a time-consuming task for data scientists. Before applying analytical methods to the data, it is essential to ensure that the data is of sufficient quality.

A previous conversion of data into a structured form is often necessary, as this representation of data is particularly suitable for further analyses. Once data is put into the desired form, erroneous, missing, or noisy data must be cleaned using data preprocessing methods. The use of different data mining methods depends on the data category. For example, not every analysis method can work with categorical data. Quantitative data may need to be normalized before modeling methods are applied.

In the modeling phase, different approaches like supervised, unsupervised, or reinforcement learning are applied to the data to find patterns, regularities, or decisions. There are many different algorithms available for this purpose.

The objective of the evaluation phase is to identify the most suitable and valid model. Before applying a model in an organization, it needs to undergo rigorous testing under controlled laboratory conditions. It's important to note that even if a model has demonstrated excellent performance in the laboratory, it does not guarantee similar performance in real-world scenarios, so continuous testing and adapting is necessary.

Deployment refers to the process of implementing the model into regular operation. To effectively utilize models in a production system, they need to be adapted regularly to match the conditions of the operating environment and seamlessly integrate with the existing infrastructure. This may entail significant costs or even the replacement of certain systems. Close collaboration between data scientists and software development teams is crucial in ensuring a successful deployment.

CRISP-DM is a very iterative and promising process, whose phases also alternate with each other. But it lacks of some necessary phases which enables to develop trustworthy AI; for example, it does not inherently take security and risk assessment into account. In 2021, M. Haakman et al. [24] tried to show that AI life cycles need to be revised, because since their inception, the challenges concerning AI have changed and new ones have arisen. After conducting interviews, Haakman et al. propose the addition of a Data collection step, a Documentation step, a Risk Assessment step, and a Monitoring step after deploying a model.

In this chapter, we would like to highlight aspects that would have to be added to the CRISP-DM in order to meet all requirements for the development as well as the deployment of safe AI applications. While safe AI has different definitions, in the context of this chapter, this term is focused on systems that pose minimal risk of failure or discrimination. Therefore, we present a blueprint for AI which, in addition to the phases of the CRSIP-DM, adds missing aspects, regarding explainability and robustness.

The blueprint consists of seven pillars that are necessary for AI systems. These seven pillars must also be taken into account by data scientists, the professionals which are responsible to develop AI systems. A data scientist combines expertise in statistics, mathematics, programming, and domain knowledge to extract valuable insights and knowledge from large and complex datasets. They use various techniques, tools, and algorithms to collect, preprocess, analyze, interpret, and model data in order to solve complex problems and make informed decisions. Additionally, they play a crucial role in developing data-driven strategies and solutions for organizations and are not infrequent responsible for designing, building, and maintaining the infrastructure and systems that enable the storage, processing, and analysis of large volumes of data. The knowledge of necessary steps to provide trustworthy AI is important.

First, in Sect. 2, we consider what influence domain-specific knowledge has on the quality of AI algorithms. In Sect. 3, we will look at the technical aspects; this includes among others the handling of data and the creation of models for machine learning. We will also look at what role structured processes play in this context. Special security aspects are described in Sect. 4, followed by a discussion of ethical aspects Sect. 5. In Sect. 6, we look at the social aspects that are significant in the development of AI. These are mainly the United Nations Sustainable Development Goals (SDG) with a specific focus on environmental aspects. Finally, Sect. 7 concludes this chapter.

2 Domain Aspects

In this section, we give a short discussion in the integration of domain experts into data analysis and discuss some important aspects that are often overlooked when challenging a data project purely based from a data scientists point of view. Since every domain has its own specific merits and peculiarities, introducing domain experts into related data science projects is of the utmost importance, especially when the results ought to be used later on in either commercial products or permanent local installments.

2.1 Integration of Domain Experts

Data scientists may lack expertise in business and domain knowledge. While data science competence can sometimes be a valuable asset in compensating for a lack

of business knowledge, it can also lead to neglecting crucial information that is needed to produce models with relevant business outcomes. Data mining processes should therefore reflect this importance. Currently, the modeling process is abstract and filters out many domain-specific factors that are essential for connecting academic research-based findings with practical, industry-focused problem-solving solutions [6]. Waller et al.[51] describe the importance of domain knowledge in the field of supply chain management. They state that domain knowledge and the analysis of data cannot be separated.

Domain experts, as the name suggests, are well-versed in understanding the data connected to their field of business and can provide valuable insights. Therefore data scientists need to work closely with such domain experts to learn from them. This is the only way to find questions that can be answered with the help of data as well as to navigate around pitfalls.

Having knowledge of innovation methods can also be beneficial for data scientists. Utilizing such techniques, for instance, can aid in identifying questions that have the potential to lead to new data-driven products, business models, or novel fields of application.

We recommend the usage of methods like data canvases [5], persona development, stakeholder interviews, or stakeholder maps [43] as tools that can help data scientists interact with domain experts. In using these tools, data scientists can reach a common ground with domain experts, which breed crucial understanding. First breakout and brainstorming sessions help to identify the required domain experts that need to be integrated, as this is often far from trivial in the setting of a larger company. Thus, the suitability and especially completeness of the people involved must always be challenged critically, especially when departments are suspected to send rather junior, and thus cheap, personal to the respective workshops.

At the same time, data must not be lost sight of. The focus is on acquiring an initial understanding of the available data, its quality, and its suitability for the possible applications at hand. This involves exploring the data, identifying its sources, understanding its structure and format, and assessing any limitations or issues.

This is important for several reasons. Firstly, it helps the data science team gain insights into the characteristics of the data they will be working with. This understanding aids in making informed decisions throughout the project, such as selecting appropriate modeling techniques and determining the feasibility of achieving the desired goals. Secondly, it might give clues for additional features, as well as new strategies for data exploitation in the context of the company. Data scientists must be weary though to not introduce a meaning and usefulness into data sets that quite isn't there, e.g., due to lack of data quality or sample selection bias.

2.2 Providing Trustworthiness and Control

Trustworthiness is considered to be a major important requirement for many data-driven products and services, as it ensures a certain compliance to principles

related to human oversight and security. Nevertheless, trustworthiness in most definitions (see [26] for the definition of the High Level Expert Group HLEG and [49] for the definition by the NIST for two of the most prominent ones) has some pitfalls, as it sometimes incorporates highly nontechnical requirements like ”societal well-being” [26] that might not coincide with functional requirements and typically requires explainability, which is a very strict and demanding feature. Thus, controllable AI [29] could be taken into consideration as an alternative, as it does not impose these requirements onto the AI system but models them rather like we model legal requirements: The operator of the system needs to be able to control that the AI does not diverge too much from the intended mode of operation (in quality of the results, as well as in the way they are achieved) and is able to circumvent the AI in cases this happens. Explainability is not required, i.e., the operator does not need to understand why an AI is or is not working as intended, as long as the overall system is resilient enough to be able to cope with the effects.

2.3 Security and Cyber Resilience

Data understanding also plays a significant role in developing secure data-driven applications. By thoroughly examining the data, potential vulnerabilities and risks can be identified early on. This allows for the implementation of necessary security measures to protect sensitive information and ensure compliance with regulations. A very important topic is cyber resilience. In this concept, related to cyber security, it is assumed that it is impossible to always defend against attacks and that some attackers will get through even the most sophisticated defenses [4]. Thus, a resilient system needs to be able to cope with successful attacks and either recover or change itself in order to be able to continue working as intended. While this is already difficult to achieve in non-AI systems, the missing explainability makes even standard procedures like penetration tests complicated in certain instances of AI, especially when considering self-changing systems like in reinforcement learning. Still, understanding the data might help in uncovering potential biases, errors, or inconsistencies that could impact the reliability and fairness of the resulting models, or, at least, allowing for more informed risk management. Furthermore, sanity checks could be devised that allow for at least some form of detection mechanisms regarding output or model manipulation. The output of such an analysis is often realized as a quality report on the data, describing all data sources and all fields in the data set, as well as information on how control is exerted upon them. Still, securing AI and providing resilient AI systems are a big challenge for many of the very popular AI techniques like deep learning or reinforcement learning and requires much more additional research, which will require knowledge on the domains involved.

2.4 Laws and Regulations

Another vital domain-specific aspect is the adherence to all sorts of norms, standards, laws, and regulations that are in place. While this is more or less standard for common regulations like the GDPR, knowledge on domain-specific regulations needs to be derived from cooperation with the respective experts. This is especially important in case of standards and best practices that are not made into law, as these are very hard to find out about from a pure outside perspective. This especially holds true, when these best practices do not apply for the industry as a whole, but only to a very specific subset of systems. Furthermore, as new systems should be designed to be as future proof as possible, upcoming regulations should be taken into account too. Following, we give a short example on some very important pieces of legislation that will come into effect in the European Union in recent years that will affect the usage of AI. Please keep in mind that this list is not comprehensive.

2.4.1 The AI Act

The most prominent example for novel regulatory development in the area of AI is the AI Act [14]. Its main target lies in providing the rules for a common market for AI-based systems. It not only provides a definition of what is considered to be AI, which had been the topic for a lot of debates, but also classifies the utilization of AI based on risks and application domains involved into four different categories: prohibited, high risk, limited risk, and minimal risk. For each risk class, and especially for high risk AI, guiding principles and rules are defined. At the time of writing this paper, some paragraphs of the AI-Act are still very much under discussion; thus, we omit discussing further details at this point.

2.4.2 The Data Act

The Data Act [16] focuses on manufacturers of smart devices and cloud providers and especially focuses on control over data generated by these devices. The idea behind the Data Act lies in increasing availability and interoperability of nonpersonal information and comes with several requirements for the developers and providers of IoT devices. Firstly, the design of the products must be done in a way to ensure simple real-time access to the collected data generated by the devices, as well as by connected ones. Furthermore, it grants new privileges for the owners of connected products, especially the right to request data holders to share data with a specific third party directly. In addition, it also grants privileges to public bodies, especially the right to request access to the data in cases of emergency.

2.4.3 The Data Governance Act

The main purpose of this act [13] lies in providing a framework for ensuring confident data sharing that can be reused easily from a technical perspective – aiming at providing an increasing amount of data of good quality in order to fuel innovation. This also demands infrastructure setup by the member states in order to facilitate this reuse and defines the duties of so-called *data intermediation services* as well as defines the concept of *data altruism*.

In addition, other software or system-related regulations like the NIS2-directive [15] might have an impact on design and use of a specific AI system, especially in case of critical infrastructures. Furthermore, even company-specific standards and best practices might be in place that require addition attention and need to be taken into consideration too.

2.5 Domain-Specific Peculiarities

Including domain experts is also very important in order to integrate AI-based systems well with existing functionality, especially considering system-specific nonfunctional requirements. As an example, industrial environments typically have long lifespans, which not only make the addition of new hardware and software difficult due to compatibility and performance issues but also can be problematic when introducing standard features for cyber resilience, which in turn makes achieving trustworthiness difficult. As an example, deep package inspection might not be introduced due to the performance overhead of the inspection which would lead to problematic delays in certain arts of an industrial complex. Even more problematic is the standard doctrine of patching, which is hard to do in many industrial environments due to problematic downtimes or the need for recertification [28]. In addition, domain knowledge is extremely important when dealing with real-world data, as this is typically tainted, i.e., there are errors inside the data, either from the data generation/retrieval process or due to problems in the subsequent handling. Thus, in any real-world application, data cleansing [25] is of the utmost importance, which not only requires a lot of domain knowledge but also has the danger of opening a plethora of legal issues [47].

3 Technical Aspects

In this section, we will discuss some selected aspects from the technical perspective, especially focusing on the CRISP-DM model for its structured approach. In addition, we focus on modeling the security and privacy relevant parts and comment on issues that need further reflection apart from the original approach.

3.1 Data Preparation

Data preparation is a vital part in any data-reliant system which is often forgotten or underestimated. This especially holds true for the data cleansing stage, which is not even mentioned in many scientific publications using data, despite the potentially large impact it can have on the actual results. This also holds true for anonymization techniques, which potentially introduce a lot of distortion, e.g., in the case of using generalization-based approaches for reaching k-anonymity [48]. More importantly, it has been shown in [46] that it is not even possible to give good estimations, or even lower/upper boundaries, on the distortion introduced, when executing machine learning algorithms on k-anonymized data sets: First, the actual data quality of the anonymization is largely depending on the actual data precision metrics in use [12], which is currently a largely underdeveloped topic. Second, even when comparing anonymization using the same metrics with different algorithms on the same data set, the distortion differs vastly. Third, cases have been identified where the subsequent machine learning algorithms performed better on anonymized data sets of lesser granularity (i.e., data sets that have been anonymized *stronger*), which is counterintuitive at first glance, but logical, if the anonymization by generalization is seen as a form of pre-clustering. In some singular cases, working on the anonymized sets yielded even better results than working on the original ones, due to outlier removal and pre-clustering effects. Contrary to these results, in many cases, the distortion of strong anonymization on the subsequent data analysis was non-negligible and introduced quite a negative effect. Thus, summarized, to correctly interpret the effects of anonymization, as an example of important data preparation techniques, requires a lot of attention, both from a technical and domain perspective.

3.2 Modeling

The modeling phase involves the application of various techniques to build and develop predictive or descriptive models based on the selected data set. During this phase, the data mining team selects the modeling techniques that are most appropriate for the project's objectives. This can include techniques such as decision trees, neural networks, regression analysis, or clustering algorithms. The selected models are then trained using the available data, allowing them to learn patterns, relationships, and dependencies within the data. Once the models are trained, they are evaluated to assess their performance and effectiveness by the use of metrics specific to the selected methods like accuracy, precision, recall, Cohen's Kappa [7], or predictive power. If the models do not meet the desired criteria, the team may revisit the data preparation phase to improve the quality or relevance of the data or adjust the modeling techniques used. The outcome of the modeling phase is a set of

reliable, validated, and optimized models that can be used for decision-making or generating predictions.

3.3 Evaluation

The evaluation phase is crucial for ensuring the reliability, effectiveness, and suitability of the models in real-world applications. It helps in identifying any shortcomings, biases, or limitations of the models and provides insights into their overall performance. Through this evaluation, organizations can make informed decisions about whether the developed models are suitable for deployment and further utilization. Based on the evaluation results, the data mining team can make informed decisions about the models. If the models meet the desired criteria, they can proceed to the deployment phase. However, if the models fall short or do not meet the project requirements, further iterations of the modeling phase may be necessary, including refining the modeling techniques, adjusting parameters, or revisiting the data preparation phase.

3.4 Deployment

During this phase, the focus shifts from model development and evaluation to the practical implementation of the models. The data mining team works closely with relevant stakeholders and technical teams to ensure a smooth deployment process. The deployment phase aims to transform the developed models into practical solutions that can provide value to the organization or end users. It ensures that the models are effectively utilized in real-world scenarios, enabling informed decision-making, process optimization, or other desired outcomes. Any evidence of bias, unfairness, or nontransparency should be eliminated at the beginning of this phase so that the models can be put into production.

Once deployed, the models need to be continuously monitored to ensure they are performing as expected. Regular monitoring helps identify any performance degradation, data drift, or changes in the model's effectiveness. Maintenance activities may include updating the models, retraining them with new data, or addressing any issues that arise.

3.5 Data Management

The topic of data management is currently often overlooked and reduced to the part of data collection and providing the AI systems with enough (high quality) data in order to generate the best possible models. Still, data management encompasses

far more tasks, some of which directly reflect on secure long-term utilization of a model.

In order to facilitate good long-term use of data and the resulting models, *data management plans (DMPs)* [10] should be put in place, though techniques like reinforcement learning require a far more advanced approach to providing transparency than provided by the typical, rather static, approaches for DMPs. A more in-depth analysis of the shortcomings of typical templates for DMPs, including a strategy on how to overcome them, can be found in [53].

Furthermore, in order to combat the problem of hidden bias in machine learning-based systems, the DMP should incorporate information on possible sources for bias inside the data, especially when these can get propagated into the final model. This, of course, requires the owner of the system to be aware of bias inside the data.

What should not be overlooked is the importance of a proper documentation. It is crucial to document the collection and selection of training data, done in collaboration with experts, as well as the modeling and evaluation processes.

4 Security Aspects

Challenges and threats to artificial intelligence are manifold. Therefore, in this subsection, we provide an overview on the current threat landscape with regard to the machine learning system itself and in combination with cybersecurity challenges of these applications.

The threat landscape for machine learning specifically can be structured in two parts: the attack surface and adversarial capabilities. The attack surface describes where, in regards to the phase of an artificial intelligence application, an adversary might want to attack. Papernot et al. [38] define two main stages consisting of *inference* and *training*, to give a general basis, applicable to most if not all machine learning systems. Qiu et al. [39] shift these stages to be *training* and *testing*, while both describe similar things; in order to give a better overview, we propose adapting these two definitions into three phases: *inference*, *training*, and *testing*.

The inference phase includes the data ingestion, the learning algorithms, and parameters of the model and the corresponding architecture. The training phase concerns itself with running training data through the target model and building the logic and the testing phase evaluates the outputs a model produces, given a specific input.

Adversarial capabilities can be defined as the knowledge an adversary has about the artificial intelligence system and the corresponding actions they can take [38, 39]. This can be thought of rather intuitively, as access to a model directly and knowledge about its inner workings, opens a lot more attack angles than having little of either. An adversary with the former may, for example, change a model's parameters or poison ingested data during the inference and training phase.

An adversary with little knowledge and access to a target model may choose to attack a model during the testing phase, with adversarial examples [57], where

original input that was previously classified correctly, for example, an image of a stop sign being classified by an autonomous car, is tampered with in order to be classified incorrectly. These changes can be rather obvious like putting a few stickers on a stop sign, but even more robust artificial intelligence models can be lead to misclassify through the introduction of noise that can be inconceivable to a human.

The strength and severity of such adversarial capabilities are closely interlinked with an adversary's means to gain access or knowledge about a system; therefore, cybersecurity considerations become a vital component when assessing the threat landscape of artificial intelligence applications. The ENISA threat landscape report [17] analyzes cybersecurity challenges with regard to artificial intelligence, which are a comprehensive source for risk identification. In the publication, 74 threats are listed and mapped to the AI life cycle and relevant AI assets, which have been categorized into data, model, actors, processes, environment/tools, and artifacts.

In its succeeding publication [18], ENISA introduces a comprehensive guide, which analyzed more than 230 references in order to survey risk factors and their relations. Building upon a mapping between threats targeting artificial intelligence, their underlying vulnerabilities, and the AI life cycle, suitable controls have been determined, which have been categorized into the domains *organizational*, *technical*, and *machine learning specific*. For controls outside artificial intelligence, the guide references widely used standards/frameworks, such as ISO 27001 or NIST SP800-53.

A similar initiative is taken by MITRE. Following the structure of MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) matrix [37], the MITRE's ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems) matrix [36] outlines threats to machine learning along the kill chain (i.e., Reconnaissance; Resource Development; Initial Access; ML Model Access; etc.). With their ATLAS Navigator toolkit tactics and techniques, MITRE provides a toolkit to link the identified techniques to threat intelligence.

The question of why an adversary might attack an AI system has to be addressed also. The number of reasons are too many to list entirely, but some examples may be the undermining of confidence, where a model could be held from deployment because a decrease in performance leads to less confidence in the models output. Another reason might be to mask certain input, for example, in network intrusion detection; a model could be meddled with to classify traffic generated by an adversary as nonintrusive. They could also go as far as damaging an organization's reputation by introducing a bias into a model that can have devastating repercussions not only for the organization in question, when becoming public, but also for people affected by the model such as credit-score estimation or artificial intelligence systems of the sort.

5 Ethical Aspects

Artificial intelligence works with real data as well as algorithms developed by real people, not machines. For this reason, it always happens that human experiences, attitudes, but also prejudices are reflected in the models created [21]. Barocas and Selbst [1] define important factors that determine how discrimination happens in data mining; we will outline the most relevant ones now.

The first factor is the definition of the target variable and the class labels. Discrimination can already occur here. If, for example, the task is to find the best employee, the question arises as to how "best" is defined. Part-time employees or employees on maternity leave are often at a disadvantage here.

The next source is the training data itself, which can be heavily biased. Classic examples are image recognition systems that have only been trained with light-skinned people. The disadvantage here is often not intentional but rooted in a lack of attentiveness or awareness on the part of the data scientists.

Especially for classical supervised machine learning models, feature selection plays an important role. It is often the case that supposedly harmless features allow conclusions to be drawn about ethnic affiliations, for example. Even if these features are important for models, they should be removed from the models in order to reduce the risk of discrimination.

Mehrabi et al.[34] presented a survey on bias and fairness in machine learning. They categorize bias on the data, algorithm, and user interaction loop. Their main categories are:

- Data to Algorithm (D2A): The bias is already in the data which is used to train the machine learning algorithm and thus might still be present in the outcome of the model.
- Algorithm to User (A2U): Although the data might be unbiased, the algorithms might introduce biases.
- User to Data (U2D): Datasets that are used to train machine learning models are very often generated by users (in contrast to, e.g., sensor data, which is produced by machines). As a consequence, inherent bias that is existent in users is transferred to the data.

Tables 1, 2, and 3 show the different manifestations of the bias for each of the categories. On the one hand, these tables serve to show practitioners the different types of bias and also what causes them. Thus, it is possible to exclude bias as far as possible in the different phases of a data science project, which is also facilitated by the structured application of processes such as CRISP-DM.

Table 1 Type and sources of data to algorithm biases according to [34]

Type	Description
Measurement bias	Bias arises from the particular features that are chosen, utilized, or measured
Omitted variable bias	Occurs when one or more important variables are left out of the model
Representation bias	Arises from how the sample is taken from a population during the data collection process
Aggregation bias	Arises when false conclusions about individuals are drawn from observing the entire population
Sampling bias	Arises due to nonrandom sampling from subgroups
Longitudinal data fallacy	Is important when analyzing temporal data
Linking bias	Arises when network attributes from user connections, activities, or interactions differ and misrepresent the true behavior of users

Table 2 Type and sources of algorithm to user biases according to [34]

Type	Description
Algorithmic bias	The bias is not present in the input data and is added purely by the algorithm. Influencing criteria are, e.g., optimization functions, regularizations, application of models on the whole data, or subgroups
User interaction bias	The user itself imposes his/her self-selected biased behavior and interaction
Popularity bias	More popular items tend to be exposed more
Emergent bias	Occurs as a result of use and interaction with real values. Influencing factors are change in population, cultural values, or societal knowledge
Evaluation bias	Happens during the model evaluation and includes the use of inappropriate benchmarks

Table 3 Type and sources of user to data biases according to [34]

Type	Description
Historical bias	Historical bias is the already existing bias and socio-technical issues in the world and can seep into from the data generation process even given a perfect sampling and feature selection
Population Bias	Arises when statistics, demographics, representatives, and user characteristics are different in the user population of the platform from the original target population
Self-selection bias	A subtype of the selection or sampling bias in which subjects of the research select themselves
Social bias	Happens when others' actions affect our judgment
Behavioral bias	Arises from different user behavior across platforms, contexts, or different datasets
Temporal bias	Arises from differences in populations and behaviors over time
Content production bias	Arises from structural, lexical, semantic, and syntactic differences in the contents generated by users

6 Social Aspects

In this section, we will discuss social aspects for this roadmap, i.e., what challenges that will not be solvable in a technical manner interact with the widespread integration of AI-based systems.

We already discussed the importance of explainability and robustness of AI applications in Sect. 5. Now we will have a look at the transfer of technical knowledge to the society.

6.1 Human in the Loop

The "human in the loop" principle in AI refers to a design or operational approach where human involvement is incorporated into an automated or AI-driven system. It involves the inclusion of human decision-making, oversight, or intervention at various stages of the AI process to enhance performance, address limitations, ensure accountability, and promote ethical considerations. [55]

In practical terms, the human in the loop principle can be implemented in different ways. For example, it can involve humans reviewing and validating AI-generated outputs, providing feedback or corrections to improve the system's accuracy, or making final decisions based on AI-generated recommendations. This human involvement helps to leverage human expertise, contextual knowledge, and ethical judgment, complementing the capabilities of artificial intelligence systems. [9]

Using a human in the loop can offer benefits for specific tasks. In certain cases, a human expert can provide valuable experience, domain knowledge, and conceptual understanding to the AI pipeline. While this may not always be the case, such approaches are not only legally sound but also crucial in many application areas where understanding the "why" is often more significant than simply achieving a classification outcome [27].

6.2 AI for Social Good

The advancement of artificial intelligence (AI) is presently primarily driven by commercial interests. However, the AI for Social Good (AI4SG) initiative aims to harness AI's potential for the benefit of society. This entails shifting the focus from solely pursuing financial gains through AI technology to considering the well-being of individuals and the environment. Consequently, it becomes crucial to establish ethical standards and criteria for the design, development, and implementation of AI. Cowls et al. [8] argue that the United Nations (UN) SDGs [50] provide a valid framework for benchmarking projects for their potential to socially good uses. The

SDGs are globally accepted targets and are therefore well suited as criteria for measuring the positive social impact of AI.

6.3 Explainability and Interpretability

Explainable AI (XAI) [41] is a scientific field dedicated to developing AI models that can provide understandable explanations for their decision-making processes. XAI focuses on generating post hoc explanations for AI models' decisions and behaviors. The goal is to provide users with a human-understandable account of why an AI system made a particular decision. This aims to bridge the gap between the complexity of AI algorithms and the need for transparency and interpretability in various domains.

Interpretable features are identified and utilized to ensure that the explanations align with human understanding and domain knowledge. XAI techniques can be categorized as post hoc or inherent explanations, where post hoc methods analyze the model's internal representations, and inherent explanations come from naturally interpretable models [56]. XAI methods often employ various techniques to extract, summarize, and visualize relevant information from the model, allowing users to understand the factors influencing the model's output [23]. These techniques include:

- Rule-based explanations: These methods aim to capture the decision-making process by generating human-readable rules. Examples include decision trees, rule lists, and production systems [22].
- Feature importance: These methods determine the contribution of individual features or variables in the model's decision. Techniques such as feature attribution and sensitivity analysis are employed to identify the most influential factors [2].
- Local explanations: Local interpretability focuses on explaining individual predictions rather than the entire model. Techniques like LIME (Local Interpretable Model-agnostic Explanations) [35, 58] generate simplified, locally faithful models to explain specific predictions [31].
- Global explanations: Global interpretability provides an overall understanding of the model's behavior. Techniques like partial dependence plots and Shapley values help identify the relationships between features and the model's output on a broader scale [31].

Explainable AI has gained significant attention in domains where transparency and interpretability are crucial and domain-specific considerations are taken into account, tailoring explanations to meet the requirements and ethical standards of different fields, such as health care, finance, and autonomous vehicles. It allows stakeholders to trust AI systems, detect biases, and identify potential vulnerabilities. Advancements in XAI involve exploring approaches like integrating human feedback [55], developing hybrid models, and investigating the trade-offs between explainability and performance.

Ultimately, the pursuit of explainable and interpretable AI aims to strike a balance between the power and complexity of AI systems and the need for human comprehension and control. By lifting the veil on the black box of AI, we can build more ethical, trustworthy, and responsible AI systems that benefit society while mitigating potential risks and biases [11, 30].

6.4 Trustworthiness

Trustworthy AI encompasses a broader set of principles and concerns. It refers to the development and deployment of AI systems that are reliable, fair, secure, and aligned with human values. Trustworthy AI includes considerations such as ethical use, accountability, privacy, robustness, and fairness. It aims to build AI systems that can be trusted by users and society at large [54].

6.5 Transparency

Transparency in AI applications refers to the degree to which the decision-making process and underlying mechanisms of an AI system are made accessible, understandable, and explainable to users and stakeholders. It involves shedding light on how the AI system arrives at its outputs, predictions, or decisions. Transparency is crucial for building trust, enabling accountability, and facilitating the responsible use of AI. It is achieved by designing AI models that reveal their inner workings, including algorithms, features, and decision rules. Therefore models should be developed with a well-defined process like CRISP-DM, as these give a standard order of operations that includes how the data was derived as well as how features may be selected. Some researchers [20] propose the concept of Transparency by Design for Artificial Intelligence applications.

7 Conclusion

Especially in recent years, as artificial intelligence has increasingly found its way into products and processes of daily life, the requirements for the use of the algorithms have also evolved. Issues of ethics and safety are particularly important when humans are directly affected, as is the case with self-driving cars. However, it is also necessary to define and adhere to principles in other applications that seem innocuous at first glance.

In this work, we have presented a set of concepts to enable a prerequisite for the secure development, application, and use of artificial intelligence. They will be successful if they are already considered in the project during the planning

phase. In the development phase, experts create models who have learned not only the technical aspects but also the ethical aspects as part of their training. Before applications are launched on the market, an intensive test phase is necessary to assess risks to users, society, and the environment. This also includes testing the applications in security-relevant aspects.

The legal aspects provide a framework for working with AI and for handling personal data. This, coupled with a high degree of transparency, ensures that the trust of both end users and society in the new technologies will also increase and that applications will also become established.

References

1. Barocas, S., Selbst, A.D.: Big data's disparate impact. In: California Law Review, pp. 671–732 (2016)
2. Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J.M., Eckersley, P.: Explainable machine learning in deployment. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 648–657 (2020)
3. Biswas, A., Kolczynska, M., Rantanen, S., Rozenshtein, P.: The role of in-group bias and balanced data: a comparison of human and machine recidivism risk predictions. In: Proceedings of the 3rd ACM SIGCAS Conference on Computing and Sustainable Societies, pp. 97–104 (2020)
4. Björck, F., Henkel, M., Stirna, J., Zdravkovic, J.: Cyber resilience–fundamentals for a definition. In: New Contributions in Information Systems and Technologies, vol. 1, pp. 311–316. Springer, Berlin (2015)
5. Brock, J., von Enzberg, I.S., Kühn, I.A., Dumitrescu, I.R.: Process mining data canvas: a method to identify data and process knowledge for data collection and preparation in process mining projects. Proc. CIRP **119**, 602–607 (2023)
6. Cao, L., Yu, P.S., Zhang, C., Zhao, Y.: Domain Driven Data Mining. Springer, Berlin (2010)
7. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Measur. **20**(1), 37–46 (1960). https://doi.org/10.1177/001316446002000104
8. Cowls, J., Tsamados, A., Taddeo, M., Floridi, L.: A definition, benchmark and database of AI for social good initiatives. Nat. Mach. Intell. **3**(2), 111–115 (2021)
9. Deng, C., Ji, X., Rainey, C., Zhang, J., Lu, W.: Integrating machine learning with human knowledge. Iscience **23**(11), 101656 (2020)
10. Donnelly, M.: Data management plans and planning. In: Managing Research Data, pp. 83–103 (2012)
11. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. Commun. ACM **63**(1), 68–77 (2019)
12. El Emam, K., Dankar, F.K., Issa, R., Jonker, E., Amyot, D., Cogo, E., Corriveau, J.P., Walker, M., Chowdhury, S., Vaillancourt, R., et al.: A globally optimal k-anonymity method for the de-identification of health data. J. Am. Med. Inform. Assoc. **16**(5), 670–682 (2009)
13. European Commission: Proposal for a Regulation of the European Parliament and of the Council on European Data Governance (Data Governance Act). European Commission (2020). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52020PC0767. COM/2020/767 final
14. European Commission: Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts. European Commission (2021). https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex:52021PC0206. Proposal for a Regulation of the European Parliament and of the Council, No. COM/2021/206 final

15. European Commission: Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive). European Commission (2022). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32022L2555&qid=1694550065818. PE/32/2022/REV/2
16. European Commission: Proposal for a Regulation of the European Parliament and of the Council on harmonised rules on fair access to and use of data (Data Act). European Commission (2022). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52022PC0068. COM/2022/68 final
17. European Union Agency for Cybersecurity (ENISA): AI Cybersecurity Challenges—Threat Landscape for Artificial Intelligence. https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges (2020). Accessed Dec 2021
18. European Union Agency for Cybersecurity (ENISA): Securing Machine Learning Algorithms. https://www.enisa.europa.eu/publications/securing-machine-learning-algorithms (2021). Accessed Mar 2022
19. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., et al.: Knowledge discovery and data mining: towards a unifying framework. In: KDD, vol. 96, pp. 82–88 (1996)
20. Felzmann, H., Fosch-Villaronga, E., Lutz, C., Tamò-Larrieux, A.: Towards transparency by design for artificial intelligence. Sci. Eng. Ethics **26**(6), 3333–3361 (2020)
21. Fu, R., Huang, Y., Singh, P.V.: Artificial intelligence and algorithmic bias: source, detection, mitigation, and implications. In: Pushing the Boundaries: Frontiers in Impactful OR/OM Research, pp. 39–63. INFORMS (2020)
22. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems (2018). arXiv preprint arXiv:1805.10820
23. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: Xai—explainable artificial intelligence. Sci. Robot. **4**(37), eaay7120 (2019)
24. Haakman, M., Cruz, L., Huijgens, H., van Deursen, A.: AI lifecycle models need to be revised. An exploratory study in Fintech (2020). arXiv preprint arXiv:2010.02716
25. Hernández, M.A., Stolfo, S.J.: Real-world data is dirty: data cleansing and the merge/purge problem. Data Mining Knowl. Discovery **2**, 9–37 (1998)
26. High-Level Expert Group on Artificial Intelligence: Ethics Guidelines for Trustworthy AI. Publications Office of the European Union, Luxembourg (2019). https://doi.org/10.2759/346720
27. Holzinger, A.: The next frontier: AI we can really trust. In: Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2021, Virtual Event, September 13–17, 2021, Proceedings, Part I, pp. 427–440. Springer, Berlin (2022)
28. Kieseberg, P., Weippl, E.: Security challenges in cyber-physical production systems. In: Software Quality: Methods and Tools for Better Software and Systems: 10th International Conference, SWQD 2018, Vienna, Austria, January 16–19, 2018, Proceedings 10, pp. 3–16. Springer, Berlin (2018)
29. Kieseberg, P., Weippl, E., Tjoa, A.M., Cabitza, F., Campagner, A., Holzinger, A.: Controllable ai-an alternative to trustworthiness in complex AI systems? In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction, pp. 1–12. Springer, Berlin (2023)
30. Li, X., Xiong, H., Li, X., Wu, X., Zhang, X., Liu, J., Bian, J., Dou, D.: Interpretable deep learning: interpretation, interpretability, trustworthiness, and beyond. Knowl. Inform. Syst. **64**(12), 3197–3234 (2022)
31. Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I.: From local explanations to global understanding with explainable AI for trees. Nat. Mach. Intell. **2**(1), 56–67 (2020)
32. McCarthy, J., Minsky, M., Rochester, N., Shannon, C.E.: A proposal for the dartmouth summer research project on artificial intelligence. http://raysolomonoff.com/dartmouth/boxa/dart564props.pdf

33. McDermott, D.: Artificial intelligence meets natural stupidity. ACM SIGART Bull. (57), 4–9 (1976)
34. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6), 1–35 (2021)
35. Mishra, S., Sturm, B.L., Dixon, S.: Local interpretable model-agnostic explanations for music content analysis. In: ISMIR, vol. 53, pp. 537–543 (2017)
36. MITRE: MITRE-ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems). https://atlas.mitre.org/. Accessed Nov 2021
37. MITRE: MITRE-ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge). https://attack.mitre.org/. Accessed Nov 2021
38. Papernot, N., McDaniel, P., Sinha, A., Wellman, M.: Towards the science of security and privacy in machine learning (2016). arXiv preprint arXiv:1611.03814
39. Qiu, S., Liu, Q., Zhou, S., Wu, C.: Review of artificial intelligence adversarial attack and defense technologies. Appl. Sci. **9**(5), 909 (2019)
40. Robinson, J.P., Livitz, G., Henon, Y., Qin, C., Fu, Y., Timoner, S.: Face recognition: too bias, or not too bias? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 0–1 (2020)
41. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.R.: Explainable AI: interpreting, explaining and visualizing deep learning, vol. 11700. Springer, Berlin (2019)
42. Samuel, A.L.: Some moral and technical consequences of automation—refutation. Science **132**(3429), 741–742 (1960)
43. Scholes, K., Johnson, G., Scholes, K.: Stakeholder mapping. In: Exploring Public Sector Strategy, p. 165 (2001)
44. Shannon, C.E.: The bandwagon. IRE Trans. Inform. Theory **2**(1), 3 (1956)
45. Shearer, C.: The crisp-DM model: the new blueprint for data mining. J. Data Warehousing **5**(4), 13–22 (2000)
46. Slijepčević, D., Henzl, M., Klausner, L.D., Dam, T., Kieseberg, P., Zeppelzauer, M.: k-anonymity in practice: how generalisation and suppression affect machine learning classifiers. Comput. Secur. **111**, 102488 (2021)
47. Stöger, K., Schneeberger, D., Kieseberg, P., Holzinger, A.: Legal aspects of data cleansing in medical AI. Comput. Law Secur. Rev. **42**, 105587 (2021)
48. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertainty, Fuzziness Knowl.-Based Syst. **10**(5), 557–570 (2002)
49. Tabassi, E.: Artificial intelligence risk management framework (AI RMF 1.0). NIST AI 100-1 (2023). https://doi.org/10.6028/NIST.AI.100-1
50. United Nations SDGs. https://sdgs.un.org/#goal_section. Accessed 30 May 2023
51. Waller, M.A., Fawcett, S.E.: Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. J. Bus. Logist. **34**(2), 77–84 (2013)
52. Wiener, N.: Some moral and technical consequences of automation: as machines learn they may develop unforeseen strategies at rates that baffle their programmers. Science **131**(3410), 1355–1358 (1960)
53. Williams, M., Bagwell, J., Zozus, M.N.: Data management plans: the missing perspective. J. Biomed. Inform. **71**, 130–142 (2017)
54. Wing, J.M.: Trustworthy AI. Commun. ACM **64**(10), 64–71 (2021)
55. Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., He, L.: A survey of human-in-the-loop for machine learning. In: Future Generation Computer Systems (2022)
56. Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., Zhu, J.: Explainable AI: a brief survey on history, research areas, approaches and challenges. In: Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8, pp. 563–574. Springer, Berlin (2019)
57. Yuan, X., He, P., Zhu, Q., Li, X.: Adversarial examples: attacks and defenses for deep learning. IEEE Trans,. Neural Netw. Learn. Syst. **30**(9), 2805–2824 (2019)
58. Zafar, M.R., Khan, N.M.: Dlime: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems (2019). arXiv preprint arXiv:1906.10263

Cybersecurity and the AI Silver Bullet

Anton Holmström, Daniel Innala Ahlmark, Johan Lugnet, Simon Andersson, and Åsa Ericson

1 Introduction

Automation and information technology were central in the era of Industry 3.0, often named the digital revolution. The technological development was driven by a vision of processes running without human interference. Along with the shift into the Industry 4.0 era, the concept of intelligent digital technologies became central, and AI entered the scene. The idea of intelligent machines has a history dating back to the 1940s. Marvin Minsky and John McCarthy coined the term *Artificial Intelligence (AI)* in 1956. At that time, they provided an operational description of the efforts in AI, *"... make machines use language, form abstractions, and concepts, solve kinds of problems now reserved for humans, and improve themselves"* [24]. The research and development of AI over time are described using the metaphors of natural seasons, i.e. spring and winter, to represent cycles of hype and periods of scepticism and criticism. As the time for natural seasons differs around the globe, so do the cycles of hype and scepticism. For example, this leads to unrealistic expectations of AI but also to an uncoordinated progress of research [29]. Nevertheless, AI can positively impact global societal problems [8].

Why is AI talked about as hype nowadays? First, media reports exaggerate the future of AI in business and work. Technological development takes time, and issues of different cultures and languages are a grand challenge [29]. Second, when the breakthrough finally happens, the distribution of digital solutions is straightforward. AI has passed the stages of technological development and is accessible to the public, for example, with the launch of DALL-E in 2021 and ChatGPT in November

A. Holmström (✉) · D. I. Ahlmark · J. Lugnet · S. Andersson · Å. Ericson
Luleå University of Technology, Luleå, Sweden
e-mail: anton.holmstrom@ltu.se; daniel.innala.ahlmark@ltu.se; johan.lugnet@ltu.se; simon.andersson@ltu.se; asa.ericson@ltu.se

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_2

2022. Yet, just because the technology is developed, it does not imply practical organisational benefits. It has been suggested that visions based on science fiction are better at driving social and technological change, because citizens think about a possible technological future (e.g. [21, 26]). Listening to the hype discourse saying the future is already here makes us believe AI is a silver bullet to every organisational problem.

Early AI developments emphasised replicating human intelligence and replacing humans in most processes and operations. Do we wish for this? From a technical point of view, AI is described as having human qualities, such as 'learning'. A typical description of AI is *"a system's ability to interpret external data correctly, to learn from such data, and to use those learnings to achieve specific goals and tasks through flexible adaptation"* [19]. However, others argue that such abilities are beyond the actual functions of the technology [22].

AI builds on historical data, and such data are often biased towards human norms; one example is when Amazon used AI in 2015 for recruitment. No women were selected among the applicants simply because none had been employed before. Recently, the UN secretary-general opened the Commission on the Status of Women conference (CSW67) by declaring that AI risks deepening existing discrimination and biases because of where, by whom, and for what purposes AI is developed. The AI Act, a proposal presented in 2021 to consider legislation on AI, put forward ethical and exclusion issues, among others [11]. Others argue against the AI Act, since it will hinder development, innovation, and entrepreneurial applications.

Several practitioners and researchers have called for reflection and dialogues on the impact of AI on society, organisations, and individuals. The standpoint depends on the observer; some see tremendous and intelligent innovations, while others are sceptical. AI introduction is thus a complex issue and cannot be seen as the introduction of technology but rather a change in work life as we know it. Organisational success in introducing AI depends on overcoming resistance to change, building trust [17], and addressing ethical and cybersecurity challenges [34].

This chapter presents a conceptual view and discusses AI's organisational and managerial challenges in cybersecurity. Within that context, we intend to contribute to considerations rather than taking standpoints for or against AI. Two behavioural science concepts inspire our reflection on the topic. One is solutionism, which describes an overbelief that technological solutions are the primary remedy but neglects how they create new problems. The other is 'It Seemed Like A Good Idea At That Time', which can be similar to the saying that the solution is seldom wrong but might not solve the right problem [25]. Thus, bear in mind that the chapter provides a base for considerations rather than solutions.

First, it will present cybersecurity from an organisational and managerial point of view. After that, two areas and contexts are described, i.e. AI in information classification and AI in incident handling. At the end of the chapter, AI features and challenges are discussed before concluding the chapter with implications to consider.

2 Organisational and Managerial Cybersecurity

As often in research, defining what is included or not in a term is an ongoing debate in many areas, so also in cybersecurity. One definition suggests that information security addresses measures taken to protect information everywhere, while cybersecurity is linked to the internet and vice versa, i.e. protecting digital information [32]. Based on that definition, it is concluded that cybersecurity and information security overlap to some extent. Contemporary organisations in developed countries are highly digitalised, making such discussions philosophical. Cybersecurity and information security concepts are pragmatically used in organisations in ways that fit the organisation's goals, e.g. we need to know which informational assets and digital systems we have, where they live, their vulnerabilities, how they can be threatened, how to protect them, and how to recover in case of an attack. Simplified, organisations must ensure correct management to prevent, for example, unauthorised access, disclosure, manipulation, or deletion of information. Cybersecurity governance means the organisation employs strategies and sets up goals and procedures. Cyber resilience means the organisation can operationalise necessary security measures, which will be embedded in the employees' practices (cyber hygiene). Cybersecurity thus permeates the whole organisation strategically, operationally, and in work behaviour, i.e. it is not only a responsibility of the IT department.

Digitalisation has connected the world and created a virtual and abstract dimension, different from how we behave in the physical one. The digitalisation of organisations has been discussed regarding privacy, safety, security, and equity, concluding that work is far more remote and hybrid than ever [9]. Digitalising the work environment and workforce has become a new normal, mainly because the COVID-19 pandemic and lockdown forced organisations to rush into the digital work environment. In 2023, organisations assess that they are more aware of cyber issues than a year ago, especially among business leaders. Still, a problem when organisations are connected and share digital data is that their boards, business leaders, and cyber leaders speak different languages. Further, business and cyber leaders agree on the increased cyber risks but differ on mitigation strategies. Business leaders believe more cross-sector regulation is needed, while cyber leaders believe employee awareness is essential [36]. The problem with different jargon is described as a difference between hearing and listening, i.e. a gap between being informed and having actionable knowledge. AI is likely to bring in similar but also additional challenges beyond the technical [38].

3 Information Classification: The Basis for Secure Organisations

Managing risk is a central activity for organisations, since they must protect their intellectual properties, or information assets, whether physical or digital. An

organisational asset can be, for example, customer data like payment and billing information, employee data and records, or authorisations and passwords for the organisation's IT systems. Considering digital assets, such as information stored in computers, databases, or systems, risk management focuses on threats and vulnerabilities [31]. Vulnerability, the weakness of a system, process, or procedure, makes it open for malicious actions, for example, manipulating or changing information or damaging or disrupting digital operations. Protecting organisational digital assets from being compromised is thus related to ensuring confidentiality, integrity, and availability [16], known as the CIA triad. Yet, security controls must protect the assets from unsanctioned use while simultaneously providing access to authorised users.

Risk management starts with identifying and classifying assets, i.e. knowing which intellectual properties the organisation has and their value for its businesses. The valuation of the assets is commonly assessed in terms of operational, financial, and reputational consequences. The classification is based on the CIA triad [3]. And then, roughly described, each identified asset is judged regarding its consequences for the organisation's stakeholders, such as customers and employees, e.g. low, moderate, high, or sometimes even catastrophic. The recommendations for classifying information are described in standards and frameworks, e.g. ISO 27002:2022, NIST Risk Management Framework (RMF), and Octave Allegro. However, adapting standards and frameworks is challenging, since the suggestions must be streamlined with specific organisational conditions. Besides the problem of adapting standards, four other issues in private and public organisations for conducting information classification have been formulated and discussed [2]. Those issues are the following:

- To decide on the level of detail for the work, e.g. if the classification should be done on a complete system or its subparts. Here it was found that the decision-making depended on weighting various roles arguments, i.e. those close to the technical details of a system versus those close to the core business. Such a decision directly impacts cybersecurity governance and the investment in security measures.
- An incomplete record of assets, e.g. since information is dynamic in organisations, the previous inventory of assets must be kept up to date. Here, the valuation became troublesome when new assets came up during the work, i.e. they did not exist in the registry from the start. Missing assets in the registry can thus be new information or information previously unidentified and exposed to vulnerabilities requiring additional investigation time and resources.
- Differences in experiences impact the judgment regarding consequences, e.g. an asset's low, high, or medium risk. Differences commonly result in disagreement and extensive discussions, and it was exemplified that it can lead to overprotecting assets. Cooperation between different roles and responsibilities, i.e. various experiences, is necessary for classification. Nevertheless, a conceding approach can result in inconsistency of classification.

- Language differences, e.g. roles use different jargon and interpretations of the same topic. The core issue here is communication, i.e. the capability to express something, listen to others, and understand someone else's perspective, even though it is expressed in terms of another function in the organisation. Dwelling on investigating terminology and meaning was explained as time-consuming and frustrating, often leading to too hasty decisions.

Contradictory, but a known difficulty is managing the extensive information in the classification work, as can be discerned in the above examples of problems. Further, in such a limited but foundational part of managing risks, as the identification and classification work is, the heterogeneity of data and data sources is troublesome. Additionally, the rate of information production in organisations is another challenge related to identification, e.g. where information is processed and stored, it might be across cooperating organisations. In the case of attacks on value chains, companies get negatively surprised at how interconnected their systems and, thus, informational assets are.

AI tools excel in managing enormous information flows, hence are found very helpful in identification and classification. AI to statistically support the valuation of assets has been used for a long time, for example, in forecasting market trends and operational risks. But for information classification, AI can, for example, swiftly find personal information in digital documents, thus identifying the data and all sources containing sensitive data in an organisation.

And AI chatbots can be used in a team debate as a 'second opinion' mitigating differences or supporting terminology investigations. AI results depend on past data, but humans make mistakes, misjudge, or make inconsistent trade-offs, as described in the information classification problems above [2]. AI tools can thus not be more accurate than the input data allows [33]. Even though using AI is promising, coherent and correct information identification and classification must be carefully and continuously accomplished to reap the benefits of AI. In turn, constant retraining and tuning of AI models are necessary.

4 Incident Handling: Securing Resilience and Recovery

A threats trend report suggests organisations expect rising cyberattacks on supply chains, disinformation campaigns, human errors in cyber-physical systems, targeted attacks using smart device data, hybrid threats, and AI abuse in the coming years [23]. Nation-states recognise that cyberspace is a fifth arena for military operations. However, also non-state actors are evident [30]. A typology of such non-state cybercriminals has been suggested by Sigholm [30]. Actors are, for example, script kiddies driven by curiosity, hacktivists motivated by political or social reasons, black-hat hackers using malware and viruses for their economic gain, espionage agents, and organised cybercriminals, all using different modus for financial gains. Cybercriminals have different motivations and targets, as they

also use various methods. Targeted organisations and individuals must be aware of and manage multiple threats, while antagonists can stay specific on their chosen modus. Organised cybercrime is capitalising on organisations by inventing new attack tactics, techniques, and approaches, making it difficult for organisations to impede such 'entrepreneurial' ingenuity. Organisations ethically follow regulations and laws, while organised cybercriminals do not bother, act quickly, and are innovative [1]. Cybersecurity is an unfair 'war ground' requiring more holistic, broader, deeper, better, and actionable knowledge as organisational capabilities.

The ever-changing threat landscape carries attacks that propagate at multiple levels and in several phases, i.e. searching for technical vulnerabilities in software or hardware, so-called synthetic attacks, and attacks aimed at social vulnerabilities, so-called semantic attacks [35]. The latter type of attack is, for example, phishing emails, taking advantage of people's gullibility, and encouraging them to click on links to an infected website. Such attacks open a backdoor to the organisation's network. Attacks like this often adapt to changing security measures and refrain from actions until the right circumstances appear. In other words, watching and waiting. The saying 'in the wild' describes how malware spreads unnoticed by the organisation as attacks could be hidden. Organisations thus often find themselves falling behind when they, after the fact, analyse fragments of an attack. This situation means organisations look at the past to predict the future [5]. Yet, to reduce the time between compromise, the detection of an attack and mitigation measures, organisations strive to implement cyberthreat intelligence [35], e.g. mechanisms, indicators of incidents, and actionable advice about existing and emerging threats [14]. Such a strategic approach involves collecting, analysing, and disseminating knowledge about cyberthreats within the organisation. Ultimately, it helps deter and defend the organisation by prioritising threats, aids in reconstruction after an attack, and tactically helps validate and prioritise indicators, patches, and alerts [12]. The threat intelligence approach supports organisations in formulating a practical and useful response plan [4].

The incident response and handling strategy include ensuring availability for all employees, so they know whom to contact when they discover an attack or suspect an incident. Such a response team should be available twenty-four-seven; thus, organisations often outsource the monitoring task per se. A response team can receive 10,000–15,000 alerts per day. Handling continuously rising incidents is stressful and demanding, and burnout and security fatigue are common [13, 20]. The requirements for an organisation's incident response staff are high regarding leadership and technical skills. Implementing staff rotation in and out of the incident response team is suggested, as well as opportunities to create and run workshops for training other employees and ensuring time off to recover [7].

The activities in incident response depend on the magnitude, but handling incidents is commonly described as a four-stage process [7]:

- Preparation. Setting up the structure and mechanisms for managing incidents, for example, encryption software, tracking systems, forensic workstations, backup devices, and cryptographic hashes, but also basic things like contact information

within and outside the organisation and a war room for communication and coordination.
- Detection and analysis. Not all incident indicators are guaranteed to be true; detection systems can also indicate false positives, which are improper signs of incidents. All incidents are not necessarily a cyberattack, e.g. crashed servers or human errors in using systems. A key task is understanding the normal behaviour of the systems and prioritising which of the numerous alerts are so-called true positives.
- Containment eradication and recovery. A central activity is making decisions, e.g. shutting down the system, disconnecting from the network, or disabling a function. The decision will impact the organisation's possibility of running its businesses; any system downtime is critical. The recovery includes manifold activities, such as restoring systems, rebuilding systems if needed, installing patches, and changing passwords.
- Post-incident activity. The central activity is to create organisational knowledge from incident handling. The team should debrief to investigate, e.g. what happened, whether the actions taken were appropriate and sufficient, how to prevent such incidents and how to detect similar incidents in the future.

Organisations reported extensive use of AI for detection and prediction than for response activities [6]. One reason for that was the established rule-based processes, yet one expectation was that AI in the future will be better at learning from generic input and hence produce broader output. This can be described as a too specific or too general result problem. The supportive tools for incident handling have some shortcomings and challenges; for example, intrusion detection prevention systems are found to be slow and have a high rate of false negatives. AI tools can be used to prioritise alerts more accurately. And AI tools are very useful in scanning, analysing, and supporting decision-making based on large statistical datasets. Further, AI can provide information relevant to security professionals for efficiently selecting security measures [37]. Still, effective and useful AI implementation in prediction, detection, and prioritisation is challenged by the varying risk management and responses each organisation deploys. Using AI tools for incident handling also challenges the response team to develop new skills, for example, when AI aids prioritising by turning off an alert. The user must understand why AI did so if trusting the tool's decision.

5 Securing Cybersecurity with AI: the Flip Side

The technological base of AI is machine learning and its related areas. AI builds on several well-known statistical methods, e.g. linear regression. Simplified, enormous data can be variously modelled to extract information supporting decision-making. AI tools are outstanding in such operations, but one important issue is for users to know how the tool reaches its result. This issue is related to the human nature

of curiosity and learning and is also a key to the social acceptance of AI [27]. Users thus need insights into how the AI tool reasons. The AI interpretable models, or explainability, are essential to understand *why* the tool reached a certain result or made a certain prediction. Yet, this also depends on whether the user needs to know the rationale as it depends on the AI's environment. It could be sufficient to get a result if it is a low-risk environment, e.g. a movie recommender system or recommendations for other products when shopping online [27]. In contrast, having a result solely may solve some parts but not the complete problem in many more complex organisational applications. Accordingly, the human decision-making process follows different rationales depending on the situation. For example, if the decision should be based on assessing several alternatives, a good explainability would be contrastive, explaining each alternative's outcome [27]. Explainability can address the model or the prediction level, as they can be technical or non-technical. Molnar suggests a future of analysing models rather than data. The latter will be the job of AI. Nevertheless, there is a need for human-friendly and tailor-made explanation models.

AI comes with great responsibilities, since the apparent benefits also bring risks, for example, introducing bias, errors that could have been prevented, and poor decision-making causing mistrust for those who should be assisted. Responsible AI is emerging in the context of AI governance, addressing ethics, morals, and values in the design, development, and deployment of AI [18]. Only some organisations have guidelines for how to use AI. Additionally, there are concerns about employees needing to understand how AI operates. A recent study has found that 11% of what employees paste into ChatGPT is sensitive data, such as patient and client information [28]. So, robustness, reliability, efficacy, privacy, and explainability can be categorised as technical concerns, while reputational, ethics, policies, and regulation are related to governance. And the main concern is that responsible AI lags behind technological progress [18].

There is certainly a dark side to AI progress. AI is not only benefitting cybersecurity but is also a tool for cybercriminals. For example, the rise in advanced disinformation campaigns using deepfake. Viewing a film where a CEO's face and voice have been altered and shares classified information from a company causes reputational damages that are hard to recover. AI can aid in designing new innovative threats and ways to carry out attacks. Also, the benefit of upscaling for organisations becomes challenging when cybercriminals use AI for malicious purposes. Increases in AI-driven cyber criminality probably result in organisations allocating a higher cybersecurity budget. Further, since cybercriminals can find vulnerabilities in other systems and manipulate data, they can do so with the AI tool. For example, altering the data sets and changing parameters causes model training faults, and criminals can thus steer the AI outputs in a certain direction [10]. So, the scale and effectiveness of AI go both ways.

6 Turning the Silver Bullet into a Silver Lining

AI has here been used in a general form to denote decision support tools. Our effort has been to look beyond the technology and discuss the organisational challenges of AI in cybersecurity. We introduced two concepts from behavioural science; one was solutionism, meaning that technology will solve every problem. Within the field of AI, another concept comes to mind, i.e. a phenomenon called the AI effect. The concept is credited to John McCarthy, meaning that when AI works successfully, no one considers it to be AI anymore [15]. This leaves organisations, society, and researchers to solve everything that is not yet done in AI. Solutionism seems to drive the discourse on the hype side, and the AI effect in terms of all the things that are not yet solved seems to drive the downside. In this one-side-or-the-other approach, we forget that AI tools already operate in many digital processes today. Organisations already use AI, for example, to scan the contents in documents, forecast market trends, recognise images, conduct various analyses, and inventory activities. However, the vision of responsible AI also suggests considering the social dimensions. Another concept we introduced from behavioural science was 'It Seemed Like A Good Idea At That Time', meaning that we often jump to solutions without reflecting on consequences, e.g. societal impacts, biases, discrimination, and exclusion. In the context of cybersecurity, we suggest considering:

- Even if it seems like a good idea at the time, ensure that AI implementation creates more advantages than disadvantages when applied to solve a problem. If in doubt, reconsider another solution. Several managers feel an urgency to introduce new technologies, but a common dilemma is that the benefits for the specific organization are often unclear. Simply not foreseeing what purposes such technology will be used for and what rebound effects it creates.
- Clarify the expectations on AI and carefully analyse its limitations and potential biases for its intended use. Figure out if the implementation may or may not create a false sense of security among employees. The media hype and the marketing of AI tools make us perceive that AI can do anything and everything. Yet, implementing new technologies must be followed up with extensive training. Employees' quick adaption to using AI as a tool has already shown that they are unaware of sharing sensitive information when asking questions to ChatGPT, for example.
- Removing the human from decision-making is a double-edged sword. Just like outsourcing tasks, it can deplete the organisation's knowledge. Consider whether the AI implementation insists on upgrading additional human skills, such as capabilities related to understanding AI models and explainability. Decision-making is a core capability when it comes to cybersecurity, whether on a C-level or the core business level. If AI tools are perceived as a 'black box' and the information base for its recommendations cannot be investigated, the organization risks being drained of core decision-making competencies.

Driving progress in AI by applying a responsible approach can turn the silver bullet hype into a silver lining for cybersecurity, thus securing organisations.

References

1. Abu, M.S., Selamat, S.R., Ariffin, A., Yusof, R.: Cyber threat intelligence–issue and challenges. Indones. J. Electr. Eng. Comput. Sci. **10**(1), 371–379 (2018)
2. Andersson, S.: Problems in information classification: insights from practice. Inform. Comput. Secur. **31**(4), 449–462 (2023). https://doi.org/10.1108/ICS-10-2022-0163
3. Bergquist, J.H., Tinet, S., Gao, S.: An information classification model for public sector organizations in Sweden: a case study of a Swedish municipality. Inform. Comput. Secur. (2021). https://doi.org/10.1108/ICS-03-2021-0032
4. Berndt, A., Ophoff, J.: Exploring the value of a cyber threat intelligence function in an organization. In: Information Security Education. Information Security in Action: 13th IFIP WG 11.8 World Conference, WISE 13, Maribor, Slovenia, September 21–23, 2020, Proceedings 13, pp. 96–109. Springer, Berlin (2020)
5. Bromiley, M.: Threat intelligence: what it is, and how to use it effectively. SANS Inst. InfoSec Reading Room **15**, 172 (2016)
6. Capgemini: Reinventing Cybersecurity with artificial intelligence: the new frontier in digital security (2019). https://www.capgemini.com/wp-content/uploads/2019/07/AI-in-Cybersecurity_Report_20190711_V06.pdf
7. Cichonski, P., Millar, T., Grance, T., Scarfone, K.: Computer security incident handling guide: recommendations of the National Institute of Standards and Technology. Tech. Rep. NIST SP 800-61r2, National Institute of Standards and Technology (2012). https://doi.org/10.6028/NIST.SP.800-61r2
8. Cowls, J., Tsamados, A., Taddeo, M., Floridi, L.: A definition, benchmark and database of AI for social good initiatives. Nat. Mach. Intell. **3** (2021). https://doi.org/10.1038/s42256-021-00296-0
9. Dick, E.: Public policy for the metaverse: key takeaways from the 2021 ar/vr policy conference. Tech. rep., Information Technology and Innovation Foundation (2021)
10. Durbin, S.: Council Post: How Criminals Use Artificial Intelligence To Fuel Cyber Attacks (2020). https://www.forbes.com/sites/forbesbusinesscouncil/2020/10/13/how-criminals-use-artificial-intelligence-to-fuel-cyber-attacks/
11. Feingold, S.: The EU's Artificial Intelligence Act, explained (2023). https://www.weforum.org/agenda/2023/03/the-european-union-s-ai-act-explained/
12. Friedman, J., Bouchard, M.: Definitive guide to cyber threat intelligence: using knowledge about adversaries to win the war against targeted attacks. CyberEdge Group (2015)
13. Furnell, S., Thomson, K.L.: Recognising and addressing 'security fatigue'. Comput. Fraud Secur. **2009**, 7–11 (2009). https://doi.org/10.1016/S1361-3723(09)70139-3
14. Gartner: Definition: Threat Intelligence (2013). https://www.gartner.com/en/documents/2487216
15. Geist, E.M.: It's already too late to stop the AI arms race—we must manage it instead. Bull. Atom. Scientists **72**, 318–321 (2016). https://doi.org/10.1080/00963402.2016.1216672
16. Gerber, M., von Solms, R.: Management of risk in the information age. Comput. Secur. **24** (2005). https://doi.org/10.1016/j.cose.2004.11.002
17. Glikson, E., Woolley, A.W.: Human trust in artificial intelligence: review of empirical research. Acad. Manag. Ann. **14**, 627–660 (2020). https://doi.org/10.5465/annals.2018.0057
18. Gulley, A.: Here's why companies should commit to responsible AI (2023). https://www.weforum.org/agenda/2023/03/why-businesses-should-commit-to-responsible-ai/

19. Haenlein, M., Kaplan, A.: A Brief history of artificial intelligence: on the past, present, and future of artificial intelligence. Calif. Manag. Rev. **61**, 5–14 (2019). https://doi.org/10.1177/0008125619864925
20. Henderikx, M., Stoffers, J.: An Exploratory literature study into digital transformation and leadership: toward future-proof middle managers. Sustainability **14** (2022). https://doi.org/10.3390/su14020687
21. Kymalainen, T.: Science Fiction prototypes as a method for discussing socio-technical issues within emerging technology research and foresight. Athens J. Technol. Eng. (2016). https://doi.org/10.30958/ajte.3-4-4
22. Marcus, G., Davis, E.: Rebooting AI: Building Artificial Intelligence We Can Trust. Vintage (2019)
23. Mattioli, R., Malatras, A., Hunter, E.N., Penso, M.G.B., Bertram, D., Neubert, I.: Identifying emerging cyber security threats and challenges for 2030 (2023). https://www.enisa.europa.eu/publications/enisa-foresight-cybersecurity-threats-for-2030
24. McCarthy, J., Minsky, M.L., Rochester, N., Shannon, C.E.: A proposal for the dartmouth summer research project on artificial intelligence, August 31, 1955. AI Mag. **27**, 12–12 (2006). https://doi.org/10.1609/aimag.v27i4.1904
25. Michie, S., Atkins, L., Gainforth, H.L.: Changing behaviour to improve clinical practice and policy. In: Novos Desafios, Novas Competências: Contributos Atuais da Psicologia. Braga: Axioma-Publicações da Faculdade de Filosofia, pp. 41–60 (2016)
26. Miller, C.A., Bennett, I.: Thinking longer term about technology: is there value in science fiction-inspired approaches to constructing futures? Sci. Public Policy **35**, 597–606 (2008). https://doi.org/10.3152/030234208X370666
27. Molnar, C.: Interpretable Machine Learning. Lulu.com (2020). Google-Books-ID: jBm3DwAAQBAJ
28. Russell, B.: Legal Expert Raises Confidentiality Concerns Over Employees Inputting Sensitive Data into ChatGPT - IFA Magazine (2023). https://ifamagazine.com/article/legal-expert-raises-confidentiality-concerns-over-employees-inputting-sensitive-data-into-chatgpt/
29. Shin, Y.: The spring of artificial intelligence in its global winter. IEEE Ann. Hist. Comput. **41**(4), 71–82 (2019). https://doi.org/10.1109/MAHC.2019.2922909
30. Sigholm, J.: Non-state actors in cyberspace operations. J. Military Stud. **4**, 1–37 (2013). https://doi.org/10.1515/jms-2016-0184
31. von Solms, B.: Information security—a multidimensional discipline. Comput. Secur. **20**, 504–508 (2001). https://doi.org/10.1016/S0167-4048(01)00608-3
32. von Solms, B., von Solms, R.: Cybersecurity and information security—what goes where? Inform. Comput. Secur. **26**, 2–9 (2018). https://doi.org/10.1108/ICS-04-2017-0025
33. Souza, R.R., Coelho, F.C., Shah, R., Connelly, M.: Using artificial intelligence to identify state secrets (2016). https://doi.org/10.48550/arXiv.1611.00356
34. Timmers, P.: Ethics of AI and cybersecurity when sovereignty is at stake. Minds Mach. **29** (2019). https://doi.org/10.1007/s11023-019-09508-4
35. Tounsi, W.: What is cyber threat intelligence and how is it evolving? In: Cyber-Vigilance and Digital Trust: Cyber Security in the Era of Cloud Computing and IoT, pp. 1–49. Wiley Online Library (2019)
36. WEF: Global Cybersecurity Outlook 2023 (2023). https://www.weforum.org/reports/global-cybersecurity-outlook-2023/
37. Wirkuttis, N., Klein, H.: Artificial intelligence in cybersecurity. In: Cyber, Intelligence, and Security, vol. 1 (2017)
38. Wisniewski, M., Gladysz, B., Ejsmont, K., Wodecki, A., Van Erp, T.: Industry 4.0 solutions impacts on critical infrastructure safety and protection–a systematic literature review. IEEE Access **10** (2022). https://doi.org/10.1109/ACCESS.2022.3195337

Artificial Intelligence and Differential Privacy: Review of Protection Estimate Models

Minna Kilpala and Tommi Kärkkäinen

1 Introduction

Personal data is everywhere and wherever you go, someone or something is asking details of the information that relates to an identified or identifiable individual. Personal data is also the currency of today: Instead of paying money for applications or service providers, we compensate their efforts by sharing our personal information with them. There are many location-based and healthcare services which do not work without personal data, and for other services such as social networks, we choose to share our individual information. The creation of user-tailored services such as online ads and recommendation systems results from the collection and combination of personal data from many sources.

It is by no means surprising that many regulatory bodies and nongovernmental organizations are concerned about privacy. Consequently, there has been an active development in the field of personal data protection in recent years. Privacy legislation has been created and updated in many countries and areas, such as GDPR (General Data Protection Regulation) in Europe [52]. To respond to the increased demand for privacy, different privacy models and their implementations have been developed [61]. However, the use of these models to protect privacy often lacks a clear framework on how well the models used with some specific configuration actually protect personally identifiable information (PII) and how well the requirements set by legislation are filled.

Privacy-preserving data analysis is a way to analyze data without compromising the rights or interests of individuals. Privacy-preserving processing can be based on many techniques and approaches. It can be divided, for example, into [53]

M. Kilpala (✉) · T. Kärkkäinen
Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
e-mail: tommi.p.karkkainen@jyu.fi

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_3

(i) cryptographic methods like Secure Multiparty Computation (SMCP) [60]; (ii) non-perturbative methods like k-anonymity [51]; and (iii) perturbative methods like Differential Privacy [15]. Data mining methods such as clustering, classification and association rule mining, unsupervised and supervised machine learning methods, and pattern recognition techniques in both descriptive and predictive form can be modified for use [36]. Privacy can be addressed and protected at different phases of the data processing lifecycle: during collection, publishing, distributing, and outputting protected data and the corresponding analysis results [36].

Differential privacy is one of the key techniques to ensure that personalized information remains non-disclosed [15–17]. Its popularity has been steadily increasing during the past decade, mainly due to the rigorous definition of privacy that results from mathematical, theoretical, and relational proofs and yields low computational costs [61]. Simply by adding suitable noise to the use of sensitive data through database queries ensures that the released information will not reveal whether an individual is contained in a database or not.

In this article, we focus on existing techniques and frameworks that are used to assess and confirm that the desired level of privacy has been reached. This is a follow-up article to our earlier work [37], which by using the umbrella literature review methodology summarized the "big picture" of DP: How has DP evolved since the original definitions, how extensively is DP used, and what are the mostly addressed application domains of DP. Our review concluded the importance of a mathematically sound privacy definition provided by DP. We also pointed out the needs for future work and efforts, especially to study how more concrete mechanisms and algorithms affect the quality of privacy solutions.

2 Differential Privacy and Attacks

Differential Privacy (DP) originates from the work of Cynthia Dwork and her colleagues [16, 17], building but modifying the conceptual foundation laid by Tore Dalenius in 1977 [15]. This foundation is in principle very clear: Nothing from an individual whose personal data is part of a database should be disclosed by population estimates or through database queries. This strictness, however, cannot be achieved as such, but, with a given probability, one can define the ε-differential privacy which guarantees that removal of an individual's data item in a database would not likely affect outputs (and any consequences of outputs) generated from the published data. More formally, at the actual privacy preserving mechanism level, one adds appropriately chosen random noise to the answer of any query function to the database. This mechanism enables us to include sensitive, personalized information in public and statistical databases, which can be used to analyze trends and changes at the level of national and international populations, e.g., on census data [1]. *Local DP* refers to the case where privacy is protected with respect to the local device that performs a query [9]. *Approximate differential privacy* means that with a small probability δ one is allowed to fail to provide DP [25].

Formally above Differential Privacy definition [15] says that: "A randomized function K gives ε-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and for all $S \subseteq$ Range(K),

$$Pr[K(D_1) \in S] \leq exp(\varepsilon) \times Pr[K(D_2) \in S]."$$

In the case of *Approximate differential privacy* [16], a small δ is added to the equation which reads as:

$$Pr[\tilde{K}(D_1) \in S] \leq exp(\varepsilon) \times Pr[\tilde{K}(D_2) \in S] + \delta,$$

where $\tilde{K}$ refers to the mechanism in relation to function K.

2.1 Attacks Against Privacy

In order to protect something, one needs to understand what kind of threats are possible. Attacks against privacy protection, similarly than in the general network security, try to break or raise doubts and concerns toward the sensitive information hiding mechanisms. For instance, one can try to alter (poison) the sensitive data collection, query messages, or the retrieved information, i.e., perform *manipulation attacks* in order to make the entire system's architecture vulnerable [9]. However, if we consider only security threats, we are likely to miss some privacy threats, and thus we need to know what kinds of threats are relevant against privacy.

In the LINDDUN methodology [13], the authors created a systematic approach to model privacy-based threats, similarly to STRIDE (Spoofing, Tamping, Repudiation, Information disclosure, Denial of service, Elevation of privilege) [41, 48] is for security. In LINDDUN, privacy threats are divided into hard—where the goal is data minimization and the user is trying to provide as little personal data as possible and trust to not breach privacy—and soft privacy—where the goal is to protect data with policies, access control and audits, and the user expects that control of personal data is lost and only trust remains. The following threats are categorized under hard privacy:

L Linkability—an attacker is able to identify if items of interest are related or not.
I Identifiability—an attacker can identify the subject.
N Non-repudiation—an attacker can gain evidence that subject knows, has done, or has said something.
D Detectability—an attacker can identify whether or not an item exists.
D Information disclosure—an attacker gets access to personal data that is not supposed to be shared.

Correspondingly, threats under the soft privacy read:

U Content unawareness—user is unaware of personal data shared in system, and thus attacker can get user's identity or inaccurate information which can further lead to wrong decisions or actions.
N Policy and consent noncompliance—there is no guarantee that system actually complies with given privacy policy, and thus personal data may be revealed without users' will.

The above threats give an idea of what kind of common-level threats there may be against privacy. Our scope is more detailed, focusing on artificial intelligence and differential privacy, so we needed to go deeper into the specific threats to them.

Adversarial machine learning studies the vulnerabilities of machine learning models and algorithms. In [23], the authors present a taxonomy for classifying attacks against online machine learning algorithms. There all privacy attacks are categorized under security violation—privacy. Privacy violation means that adversary gets information from learner and security or privacy of system's user is breached. They demand that privacy-preserving learning should protect against violation by exploratory or causative attacks (under Influence main category). Exploratory attacks try to discover information of training process or training data, and causative attacks try to influence training data and thus alter training process. They present differential privacy as a strong guarantee of privacy. Biggio and Roli [4] mentions two main threats to learning algorithms: evasion attacks (manipulating input data) and poisoning attacks (misleading training algorithm).

In [40, 45], privacy attacks are categorized in four types: (i) membership inference (if the sample was part of a training set), (ii) reconstruction (recreate training sample(s) or label(s)), (iii) property inference (extracting properties that were not part of the training task, so learning something unintentionally), and (iv) model extraction (trying to build a similarly behaved model) attacks. *Floating-point attacks* refer to techniques in which one tries to retrieve information on the noise distribution used to perform DP [25]. For instance, if an adversary is able to detect the time when a sample is drawn, then information on the magnitude of noise is revealed. On the level of machine learning models, there are many points to address an attack which essentially tries to steal the private model [59]: recovery of instances of training data, inversion of a model to reveal certain features, or intelligent queries to identify part of the model's structure, etc.

In conclusion, there seems to be no common approach on how to classify attacks related to privacy in machine learning. To categorize the findings later in this study, we classified the attacks into the following groups:

1. **Interference attacks** try to manipulate learning algorithm somehow. This can connect to any of the above threats depending on what the attacker was able to manipulate. Another possible consequence is that the algorithm may start giving false results.
2. **Membership inference attacks** try to reidentify person from data set. This connects to identifiability threat.

3. **Reconstruction attacks** try to identify parameters of a specific sample. This connects to detectability, information disclosure, and non-repudiation threats as this can reveal values in data.
4. **Property inference attacks** try to learn something new. This connects closest to linkability and information disclosure threats as this may reveal unplanned properties.
5. **Model extraction attacks** try to copy the model. This can connect to any threats above and loss to intellectual property.

3 Privacy Metrics and Challenges

Privacy metrics and measures are still an evolving research area, where there is no generally approved single framework for measuring the level of privacy. However, there are already legislation that demand appropriate level of security for personal data processing, such as the European Union's GDPR (General Data Protection Regulation) [18] and California's CCPA (California Consumer Privacy Act) [6]. It would be largely beneficial to all stakeholders if there was a clear privacy measurement framework with instructions on how to ensure the correct level of privacy in different cases. In a best-case scenario, we would have clear metrics that could be used to plan, implement, and test privacy level of different privacy-enhancing technologies (PET).

Personally identifiable information (PII) or personal data as defined by GDPR ("any data that can directly or indirectly identify natural person") has already spread widely and will continue to do so. To prevent harm done to individuals or society, the use of personal data needs to be harnessed. We need personal data protection that corresponds to risks, and we need to be able to prove that it works.

In [57], privacy metrics were classified into four characteristics: (i) adversary goals, (ii) adversary capabilities, (iii) data sources, and (iv) inputs for computation metrics. In addition, the output measures referred to what kind of property does a specific privacy metric measure. Wagner and Eckhoff [57] were able to find over 80 different privacy metrics from literature, so it is no wonder that measuring privacy is considered a difficult task. The found metrics were grouped under eight groups: (1) uncertainty, (2) information gain or loss (as the amount of information that can be gained by adversary), (3) data similarity (properties of published data, like k-anonymity), (4) indistinguishability (as if adversary can distinguish between two items), (5) adversary's success probability (like success rate), (6) error (as error an adversary makes in his estimate), (7) time (as time needed to compromise privacy), and (8) accuracy/precision. These metrics will be used later in our work.

Another identification of privacy metrics was given in [42], in the context of location privacy. There, three different categories were identified: (1) formal guarantee like differential privacy and k-anonymity, (2) data distortion metrics comparing data before and after, and (3) attack correctness in connection to

mentioned adversary models and threats (points of interest and semantics, social relationships, reidentification, future mobility prediction).

Performance and utility metrics are excluded from this review. There is already plenty of materials on those, and they are usually main consideration when proposing new privacy models or PETs. For example, 76 results were left out of this review, because they were presenting new DP implementation and one common prize was that the presented model was better in performance or utility, and 9 results were left out, because they were just measuring accuracy, utility, or performance. The view in this review is that measurement of a privacy level should be an integral part of all those.

3.1 Privacy Challenges in Data Releases, Machine Learning, and Artificial Intelligence

Descriptive releases of public data, social networks, and machine learning and artificial intelligence systems collect personal data and publish it in different formats. For an individual, even with GDPR, it becomes impossible to follow all of his/her personal data and recognize what kind of risk level is involved in its use. The responsibility of protecting personal data should not be left to individuals, but instead, each data collector and beneficiary should be responsible for keeping the requested and agreed level of privacy.

So far, different anonymization techniques are commonly used, for example, when releasing statistics, but there are already proofs of failures [39]. For example, Sweeney found in [50] that with only three attributes (5-digit ZIP, gender, and date of birth) 87% of the US population would be uniquely identified. Reidentification is a big risk unless all possible aspects are considered when planning the anonymization.

Another challenge with many anonymization techniques, including differential privacy, is that they have some parameters, like ε in DP, that needs to be tuned. Selecting the correct value is usually left to people implementing PET, and it is not an easy task. There aren't any common guidelines yet, and it's difficult to assess privacy loss with given value.

An increasing risk is the possibility of connecting anonymized data with public data sources using artificial intelligence. As referred in [12], an example of such a risk in relation to a legal proceeding was given in [56]. It is not enough to handle one data release, but one should consider all possible contexts where that data could be utilized. One approach for this purpose would be to define a strong theoretical proof, like in differential privacy, but this does not yet tell anything about how well the privacy insurance was implemented or how well it will work in real life. It is therefore, already in the planning phase, important to identify what kind of proof or assurance different models give. One should be certain that all relevant aspects of privacy have been considered and then ensure using real measurements that an implemented PET really achieves the targeted level of privacy.

4 Literature Review

The aim of this article is to find out what kind of measures are available to ensure that an intended level of privacy protection is achieved in real-life implementations. Focus is on searching methods, frameworks, best practices, or any other systematic models to estimate and ensure protection. Research questions for this review were:

RQ1: How can we ensure that the expected level of protection, calculated based on the definition of differential privacy, is reached by an implementation?
RQ2: Are there any frameworks that would provide a baseline for evaluating privacy protection levels?
RQ3: Is there a way to compare these privacy protection levels against different privacy legislation such as GDPR and CCPA?

4.1 Search and Selection Steps

The first step after formulating the research questions was to define the search repository and queries. Based on our previous experience in [37], where the Scopus repository provided the utterly highest number of hits, it was decided to use it in this review. Defining search queries was a bit more challenging, because many search sentences tended to be made of either too common words, thus giving huge amount of not relevant hits, or of too rare words ending up in none or very few results. We though identified also many articles for our intended scope, so the following search strings were finally used:

SS1: TITLE-ABS-KEY ("differential privacy" AND protection AND (measure OR proof)) AND (LIMIT-TO(LANGUAGE,"English"))
SS2: TITLE-ABS-KEY ("differential privacy" AND "personal data" AND protection) AND (LIMIT-TO(LANGUAGE,"English"))

The first exclusion criterion, the English language, was already included in the search strings. The string SS1 provided 130 results and SS2 61, of which 12 were duplicates, which were removed. The results also included six of the Conference Review Document Type, which were excluded. After the basic search and the first set of exclusion criteria, there were 173 articles remaining.

In the second step, each result was briefly reviewed. The final decision on the inclusion/exclusion of an article was based on the title, abstract, and, if needed, brief look at the content of the article. The main inclusion/exclusion criterion in this step was whether an article included some kind of approach to estimate the level of privacy. Articles that were just presenting a new model, mechanism, method, framework, or algorithm based on differential privacy and usually comparing theoretically different epsilon values to the utility values did not have the aspect that was searched for. This alone already excluded 76 articles. If the article also had other privacy measures or metrics, it was included. Neither articles that had just theoretical

Table 1 Excluded article descriptions and numbers in step 2

Article type of excluded articles	Amount
Demonstration, description, overview and likes	12
DP as proof or compared to DP	6
DP not found or article not found	9
Just accuracy, utility, or performance measure	9
New usage, algorithm, model, method, mechanism, or likes of DP	76
Only theoretical model or proof	15
Other (just one of each)	6
Total	133

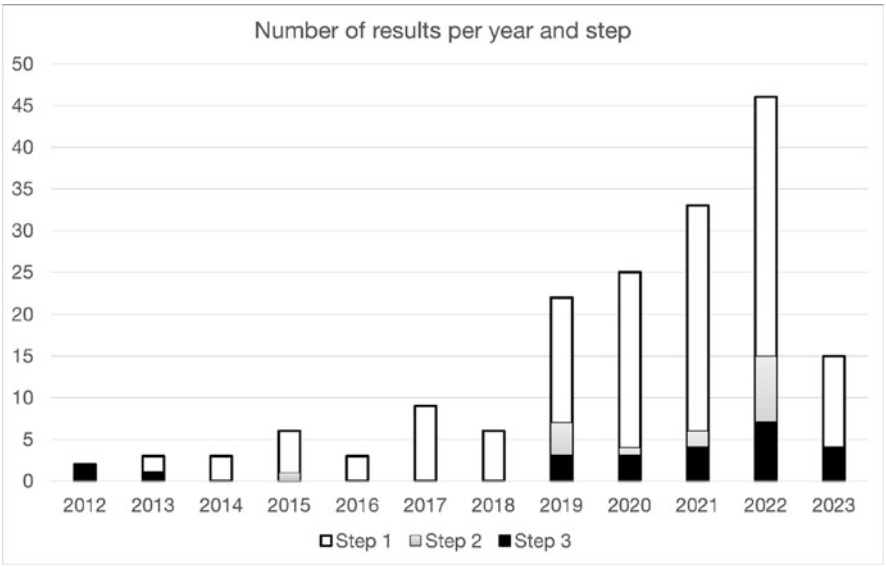


Fig. 1 Number of results included in each step by year

proofs were included as we were interested in situation after implementation. A total of 133 results were excluded in step 2, because they did not involve privacy measurement or metric, so after the second step, there were 40 articles left. In Table 1, the number of excluded results is shown by the description of the high-level exclusion reason of the article. In the third step, the resulting articles were studied, and different privacy-related characteristics were collected from them. At this point, a set of 16 articles were excluded, because they were either (i) just modifying or presenting a new DP model without any other privacy measure, (ii) they did not include DP, or (iii) they were not primary works, but just categorizing other articles. Finally, there were 24 articles to review to find answers to the research questions.

Figures 1 and 2 illustrate how the number of results changed at each step per year (1) and per type of document (2). Figure 1 shows that interest in this area is

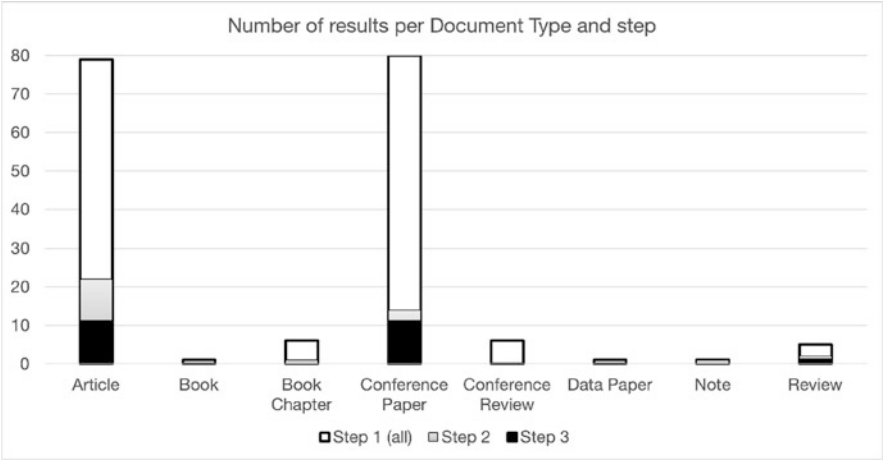


Fig. 2 Number of results included in each step by document type

still growing. The precise information on the included articles is given at the end of the article in two tables: All included articles are listed in Table 6 where ID in this review, authors, title, year, and reference are given; in Table 7, ID of each article is assigned to the source where it was published.

4.2 Result Characteristics

The final results included in the review presented a diverse and fragmented area under privacy. Differential privacy models ranged from basic central models defined by Dwork [17] to different versions of Local Differential Privacy [14, 27] and to specialized solutions like Geo-Indistinguishability [2] for location privacy. The more strict definition of privacy, ε-DP [15], was used in 15 cases, the approximate version (ε-δ)-DP [16] in 7 cases, and both were mentioned in one case. In Fig. 3, the keywords (or index terms) of the reviewed articles are presented as a word cloud.

In Table 2, the results are classified according to the kind of data release they discussed. The wide distribution indicates that many different areas are involved. However, it was a bit surprising that even if the health data obtained the highest number of release hits (3), it was still quite low. Considering how sensitive and how many legal restrictions there are for personal data related to health, the expectation was that it would have greater coverage in this review. None of the identified data release areas received an especially large number of hits; all amounts were between 1 and 3.

Of the data sets used in the cases, the different UCI data sets [33] were the most popular with six hits. Many other data sets were also mentioned, and in eight cases, no data sets were used.

Fig. 3 Word cloud of key words of the analyzed articles

Table 2 Classification of results based on data release type

Release area	Any	Classification release	Financial queries	Image data release	IoT data sharing	Legal document release	Location sharing	machine learning systems
Amount	1	2	1	1	1	1	3	2
Release area	Marketplace for data sharing	None	Query-based release	Randomized response	Restricted health data release	Social networks	Statistical release	Grand total
Amount	1	1	2	1	3	1	3	24

5 Privacy Protection Models

5.1 Privacy Attacks

Privacy attacks were collected and classified from the results. Attack types were taken as they were identified from articles and related to attack categories defined in Sect. 2.1. The identified attacks are linked to attack categories in Table 3. Each article had one to three attack types, and there was a lot of variation in the original descriptions. As can be expected when talking about privacy, the reidentification

Table 3 Number of attack categories found in 24 articles in review

Attack category	Identified attacks	Amount
Interference	Data poisoning attack	1
Membership inference	Differencing, re-identification, singling out, deniability, identity breach	8
Interference and membership inference	Poisoning, data leaks	1
Reconstruction	Attribute inference, graph structure and degree, detection of speeding vehicles, location, homogeneity, privacy leaks	7
Membership inference and reconstruction	Disclosure attack, insider attack	7
Grand total		24

attack was the most common type. However, clear challenges in the classification were revealed. Namely, the identified attack types were on different levels, for example, disclosure attack vs. detection of speeding vehicle, and partly different terminology was used in their depictions.

5.2 Privacy Metrics and Measurement Models

None of the results in the review contained any kind of common framework or model to measure privacy. The most common reference on this was legislation: GDPR was mentioned in five articles, HIPAA (US Health Insurance Portability and Accountability Act) [54] twice, and FERPA (US Family Educational Rights and Privacy Act) [55] once. It is very clear that this area needs a common framework, which could be referenced to the corresponding PET privacy level.

As [57] also points out, a common consensus on generally approved privacy metrics is still missing. Understanding the context and environment in which PET will be used is an important starting point to define the correct metrics. In the additional material in [57], nine questions were given to help with the selection of a metric: (1) Suitable Output Measures? (2) Adversary Models? (3) Data Source? (4) Availability of Input Data? (5) Target Audience? (6) Related Work? (7) Quality of Metrics? (8) Metric Implementations? and (9) Metric Parameters?. One special note related to the selection of parameters that they presented, coinciding with our findings, was that there is not even much information available on how to select ε for DP. During our selection steps, we had in the first step two articles of which one [49] was excluded as it dealt with performance or utility related to different εvalues and one was included in the review [3].

If we consider our findings in relation to the taxonomy of privacy metrics as defined in [57] and listed in Sect. 3, recalling our focus on DP, all of our results could fall under the Indistinguishability group as all DP types are mapped there. However, we found that our results could be mapped under different metrics. In

Table 4 Metric types connected to article IDs

Metric type	IDs
Adversary's success probability	1, 6, 9, 16, 21
Data similarity	11
Error	3,13
Fairness	4, 22
GDPR	2, 5, 7, 20
Indistinguishability	5, 8, 12, 18, 20, 24
Information gain or loss	10, 14, 15, 19
Time or cost	17, 23
Uncertainty	8

Table 4, we have a bit modified their version of metrics. We have added the Cost in Time metric as we found it corresponding to Time: calculating the performance cost of the needed protection level [47] or minimizing the objective (cost) function of attacker [32].

In addition, we identified two more categories for metrics: GDPR and fairness. First one might eventually be some kind on umbrella metric where needed privacy level depends on risk for individual and combines other metrics based on context and risk. The second one is an ethically important aspect, which should get much more attention. The offered protection level should be the same regardless of, for example, if individual belongs to some special, protected group (one definition is example of special categories in GDPR) or not.

Considering that all these were found with connection to DP, there is definitely need for a general model that would help to understand which metrics are relevant to which case and what level of privacy is needed.

5.3 Connecting Attacks and Metrics

In Fig. 4, the metric categories are mapped to the attack categories. Even if the attacks were just identified from a few of the reviewed articles, it is obvious that different metrics and metrics combinations are needed for defensing against different attacks. When comparing metric categories to attack categories, it seems obvious that most of the metrics are relevant for each attack. So, instead of measuring just one or two issues, we should have a full set of metrics to be considered in all cases. In addition, two attack categories did not map to any metrics. According to [21], the attack that should be considered in relation to GDPR is the identity disclosure attack, but it was not directly identified in our results. Instead, most of the identified attacks can result in identity disclosure.

To help structuring metrics, one could start by mapping metrics according to specific legislation. For example, GDPR [18] sets rules on personal data processing: "appropriate technical and organisational measures to ensure a level of security

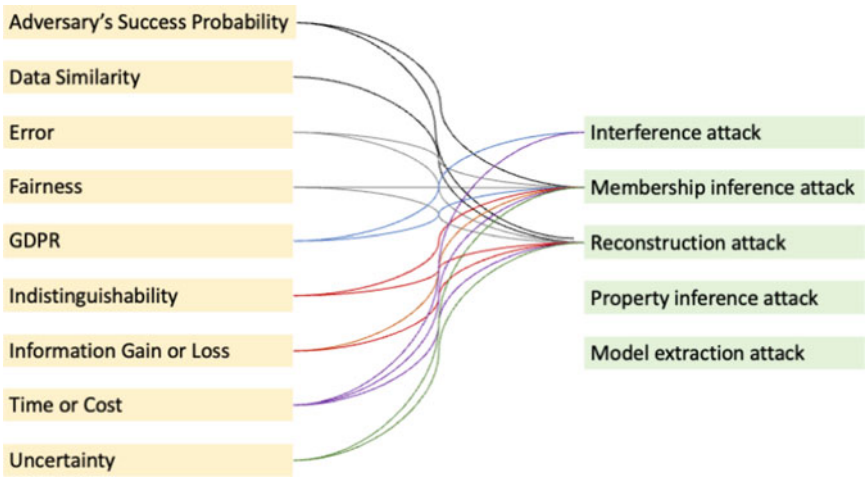


Fig. 4 Mapping metric categories (on the left) with attack categories (on the right)

appropriate to the risk." To be able to decide on the correct level of security, we need to understand what kind of risks are involved. To be able to understand risk, we need to understand what kind of threats are possible. So, we are actually returning to threat modeling. The attacks identified here are real threats to privacy in relation to machine learning. An approach could be to see how well the model is protected from specific threats. So, for example, if we think about an interference attack, we should identify all places where it can happen and measure how safe our solution is. For metrics, we could measure the success probability and the time required to breach the system from our current list. In addition, we could also measure what kind of problems different interference could cause, like how much training data can be polluted before it affects the model. Finally, we could define some values to specify which are the allowed limits in each to confirm that the actual level of privacy protection of the solution is what we were looking for.

As all metric categories in [57] are grouped output measures from single metrics, which also appear to be based on different privacy technologies, it makes the use of PETs challenging. One should combine several techniques to be able to protect personal data, but that is by no means practical. To be able to select between different PETs, we need privacy measures that can be applied to all of them.

6 Conclusions

Reflecting on our research questions, as posed in the beginning of Sect. 4, we found several different approaches to estimate the protection level in relation to RQ1. The decision of what to measure and how to measure was usually made by the authors.

There were no commonly shared and agreed principles, so deciding and ensuring the expected level of protection was made on a case-by-case basis. Neither did we find any common test model that would ensure that theoretical privacy protection was really achieved in an implementation. There were some individual approaches such as QuerySnout [11] to help identify vulnerabilities in query-based systems, and a specific attack test for certain DP implementations [25], but a systematic approach to the whole scope was lacking. Hence, the answer to RQ2 which follows from these conclusions is "No." We were unable to find any framework that could be used to ensure a correct level of privacy protection.

The LINDDUN methodology [13] was one step in the right direction, as it gives practitioners instructions on what types of privacy threats should be investigated, their details, and how to map them to PETs. However, the methodology does not go into details on what level of privacy protection should be selected. This could be in the risk assessment phase, which was outside the scope of the methodology. The article did not address machine learning-specific threats or recognize DP yet.

The answer to RQ3 was also inconclusive: we found something in that direction, but a clear, uniform way connecting a privacy framework and the legislation is still missing. In some cases, GDPR was used as a privacy metric. Another aspect to consider with legislation is that many models have parameters which affect the protection level and there is not yet any clear guidance on how those should be selected.

In the future, especially with the increasing amount of artificial intelligence, it will be important that privacy metrics consider the full context of use of personal data. DP is one of the techniques on the right track, because its definition says that nothing more should be learned of a person regardless if s/he was or wasn't included in the data. However, when DP is implemented as part of even more and more complex systems, this statement must be proved in the whole context and also after the implementation. DP also leaves open the question of what it is possible to learn about a person from the system regardless of his/her presence, nor does it approach the question of several separate systems running and impacting each other's privacy level. In addition, it may be vulnerable to a reconstruction attack, especially if the attacker can gain information on the perturbation levels used.

Considering the different categories of metrics found in this review, it turned out to be very clear that more research is needed to define privacy measurement models that cover a wide enough area. It appears that several metrics are likely to be combined or used to ensure the security of personally identifiable information. To be able to clearly answer the needs of people whose personal data is processed, authorities, and system developers, a framework that connects all the parts discussed above should be developed. Such a framework needs to cover all aspects related to privacy in artificial intelligence, be connected to other privacy and security frameworks and legislation, be clear enough to easily use and test, and give instructions on how to configure selected models. A good start would be to structure and create a common and shared model of privacy threats and privacy protection requirements in the context of artificial intelligence.

Disclaimer As artificial intelligence is part of this article, Scispace [43] was used to summarize the introduction of each included article. It was not used to create any content in this review.

List of Abbreviations

See Table 5.

Appendix

See Tables 6 and 7.

Table 5 List of abbreviations

AI	Artificial Intelligence
CCPA	California Consumer Privacy Act
DP	Differential Privacy
FERPA	Family Educational Rights and Privacy Act
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
IoT	Internet of Things
PET	Privacy Enchanting Technology
PII	Personally Identifiable Information
SMCP	Secure Multi-party Computation
SS	Search String
UCI	University of California, Irvine

Table 6 Articles included in review

ID	Authors	Title	Year	Ref
1	Ashena N. et al.	Understanding ε for differential privacy in differencing attack scenarios	2021	[3]
2	Brauneck A. et al.	Federated machine learning, privacy-enhancing technologies, and data protection laws & in medical research: scoping review	2023	[5]
3	Cerf S. et al.	Privacy protection control for mobile apps users	2023	[7]
4	Chester A. et al.	Balancing utility and fairness against privacy in medical data	2020	[8]

(continued)

Table 6 (continued)

5	Cohen A. and Nissim K.	Towards formalizing the GDPR's notion of singling out	2020	[10]
6	Cretu A.-M. et al.	QuerySnout: automating the discovery of attribute inference attacks against query-based systems	2022	[11]
7	Csányi G.M. et al.	Challenges and open problems of legal document anonymization	2021	[12]
8	Feyisetan O. et al.	Leveraging hierarchical representations for preserving privacy and utility in text	2019	[19]
9	Fotiou N. et al.	A privacy-preserving statistics marketplace using local differential privacy and blockchain: An application to smart-grid measurements sharing	2021	[20]
10	Hotz V.J. et al.	Balancing data privacy and usability in the federal statistical system	2022	[22]
11	Huang H. et al.	Privacy-preserving approach PBCN in social network with differential privacy	2020	[24]
12	Kargl F. et al.	Differential privacy in intelligent transportation systems	2013	[26]
13	Li M. et al.	Quantifying location privacy for navigation services in sustainable vehicular networks	2022	[28]
14	Liu F. and Zhao X.	Disclosure risk from homogeneity attack in differentially privately sanitized frequency distribution	2022	[30]
15	Liu F. and Zhao X.	Disclosure risk from homogeneity attack in differentially private release of frequency distribution	2022	[29]
16	Liu J. et al.	Mutual-supervised federated learning and blockchain-based IoT data sharing	2022	[31]
17	Ma Y. et al.	Data poisoning against differentially-private learners: attacks and defenses	2019	[32]
18	Matthews G.J. and Harel O.	Assessing the privacy of randomized vector-valued queries to a database using the area under the receiver operating characteristic curve	2012	[34]
19	McClure D. and Reiter J.P.	Differential privacy and statistical disclosure risk measures: an investigation with binary synthetic data	2012	[35]
20	Oguri H.	A method of decreasing connectability of derived data, using local differential privacy	2019	[38]
21	Reilly D. and Fan L.	A comparative evaluation of differentially private image obfuscation	2021	[44]
22	Salas J. et al.	Towards measuring fairness for local differential privacy	2023	[46]
23	Shen A. et al.	Exploring the relationship between privacy and utility in mobile health: algorithm development and validation via simulations of federated learning, differential privacy, and external attacks	2023	[47]
24	Wang Y.-R. and Tsai Y.-C.	The protection of data sharing for privacy in financial vision	2022	[58]

Table 7 Sources, by article ID, of included articles in review

ID	Source title
1	Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST
2	Journal of Medical Internet Research
3	Control Engineering Practice
4	2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020
5	Proceedings of the National Academy of Sciences of the United States of America
6	Proceedings of the ACM Conference on Computer and Communications Security
7	Symmetry
8	Proceedings—IEEE International Conference on Data Mining, ICDM
9	Blockchain: Research and Applications
10	Proceedings of the National Academy of Sciences of the United States of America
11	IEEE Transactions on Network and Service Management
12	WiSec 2013—Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks
13	IEEE Transactions on Green Communications and Networking
14	IEEE Transactions on Dependable and Secure Computing
15	CODASPY 2022—Proceedings of the 12th ACM Conference on Data and Application Security and Privacy
16	Security and Communication Networks
17	IJCAI International Joint Conference on Artificial Intelligence
18	Health Services and Outcomes Research Methodology
19	Transactions on Data Privacy
20	Proceedings of the 11th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2019
21	Proceedings—2021 3rd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications, TPS-ISA 2021
22	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
23	Journal of Medical Internet Research
24	Applied Sciences (Switzerland)

References

1. Abowd, J., Ashmead, R., Cumings-Menon, R., Garfinkel, S., Heineck, M., Heiss, C., Johns, R., Kifer, D., Leclerc, P., Machanavajjhala, A., Moran, B., Sexton, W., Spence, M., Zhuravlev, P.: The 2020 census disclosure avoidance system topdown algorithm. Harvard Data Science Review (Special Issue 2) (2022). https://doi.org/10.1162/99608f92.529e3cb9
2. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability. In: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13, pp. 901–914. ACM Press, New York (2013). https://doi.org/10.1145/2508859.2516735
3. Ashena, N., Dell'Aglio, D., Bernstein, A.: Understanding ϵ for Differential Privacy in Differencing Attack Scenarios. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, vol. 398 LNICST, pp. 187–206 (2021). https://doi.org/10.1007/978-3-030-90019-9_10. https://www.scopus.com/inward/

record.uri?eid=2-s2.0-85120047890&doi=10.1007%2f978-3-030-90019-9_10&partnerID=40&md5=84a2d1bcf4d0ebb03b07da6b3dd4f8d5
4. Biggio, B., Roli, F.: Wild patterns. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 2154–2156. ACM, New York (2018). https://doi.org/10.1145/3243734.3264418
5. Brauneck, A., Schmalhorst, L., Kazemi Majdabadi, M.M., Bakhtiari, M., Völker, U., Baumbach, J., Baumbach, L., Buchholtz, G.: Federated machine learning, privacy-enhancing technologies, and data protection laws in medical research: scoping review. J. Med. Int. Res. **25**, e41588 (2023). https://doi.org/10.2196/41588
6. California: California Consumer Privacy Act of 2018 (2018). https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5
7. Cerf, S., Robu, B., Marchand, N., Bouchenak, S.: Privacy protection control for mobile apps users. Control Eng. Practice **134**, 105456 (2023). https://doi.org/10.1016/j.conengprac.2023.105456
8. Chester, A., Koh, Y.S., Wicker, J., Sun, Q., Lee, J.: Balancing utility and fairness against privacy in medical data. In: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1226–1233. IEEE, Piscataway (2020). https://doi.org/10.1109/SSCI47803.2020.9308226
9. Cheu, A., Smith, A., Ullman, J.: Manipulation attacks in local differential privacy. In: 2021 IEEE Symposium on Security and Privacy (SP), pp. 883–900. IEEE, Piscataway (2021). https://doi.org/10.1109/SP40001.2021.00001
10. Cohen, A., Nissim, K.: Towards formalizing the GDPR's notion of singling out. Proc. Natl. Acad. Sci. **117**(15), 8344–8352 (2020). https://doi.org/10.1073/pnas.1914598117
11. Cretu, A.M., Houssiau, F., Cully, A., de Montjoye, Y.A.: QuerySnout. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, pp. 623–637. ACM, New York (2022). https://doi.org/10.1145/3548606.3560581
12. Csányi, G.M., Nagy, D., Vági, R., Vadász, J.P., Orosz, T.: Challenges and open problems of legal document anonymization. Symmetry **13**(8), 1490 (2021). https://doi.org/10.3390/sym13081490
13. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirem. Eng. **16**(1), 3–32 (2011). https://doi.org/10.1007/s00766-010-0115-7
14. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, pp. 429–438. IEEE, Piscataway (2013). https://doi.org/10.1109/FOCS.2013.53
15. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming, pp. 1–12. Springer, Berlin (2006)
16. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) Advances in Cryptology - EUROCRYPT 2006, pp. 486–503. Springer, Berlin (2006)
17. Dwork, C.: Differential privacy: A survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) Theory and Applications of Models of Computation, pp. 1–19. Springer, Berlin (2008)
18. European Union: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (2016). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679
19. Feyisetan, O., Diethe, T., Drake, T.: Leveraging hierarchical representations for preserving privacy and utility in text. In: 2019 IEEE International Conference on Data Mining (ICDM), pp. 210–219. IEEE, Piscataway (2019). https://doi.org/10.1109/ICDM.2019.00031
20. Fotiou, N., Pittaras, I., Siris, V.A., Polyzos, G.C., Anton, P.: A privacy-preserving statistics marketplace using local differential privacy and blockchain: an application to smart-grid measurements sharing. Blockchain Res. Appl. **2**(1), 100022 (2021). https://doi.org/10.1016/j.bcra.2021.100022
21. Hölzel, J.: Differential privacy and the GDPR. Eur. Data Protect. Law Rev. **5**(2), 184–196 (2019). https://doi.org/10.21552/edpl/2019/2/8

22. Hotz, V.J., Bollinger, C.R., Komarova, T., Manski, C.F., Moffitt, R.A., Nekipelov, D., Sojourner, A., Spencer, B.D.: Balancing data privacy and usability in the federal statistical system. Proc. Natl. Acad. Sci. **119**(31) (2022). https://doi.org/10.1073/pnas.2104906119
23. Huang, L., Joseph, A.D., Nelson, B., Rubinstein, B.I., Tygar, J.D.: Adversarial machine learning. In: Proceedings of the 4th ACM workshop on Security and artificial intelligence, pp. 43–58. ACM, New York (2011). https://doi.org/10.1145/2046684.2046692
24. Huang, H., Zhang, D., Xiao, F., Wang, K., Gu, J., Wang, R.: Privacy-preserving approach PBCN in social network with differential privacy. IEEE Trans. Netw. Serv. Manag. **17**(2), 931–945 (2020). https://doi.org/10.1109/TNSM.2020.2982555
25. Jin, J., McMurtry, E., Rubinstein, B.I.P., Ohrimenko, O.: Are we there yet? Timing and floating-point attacks on differential privacy systems. In: 2022 IEEE Symposium on Security and Privacy (SP), pp. 473–488. IEEE, Piscataway (2022). https://doi.org/10.1109/SP46214.2022.9833672
26. Kargl, F., Friedman, A., Boreli, R.: Differential privacy in intelligent transportation systems. In: Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 107–112. ACM, New York (2013). https://doi.org/10.1145/2462096.2462114
27. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? SIAM J. Comput. **40**(3), 793–826 (2011). https://doi.org/10.1137/090756090
28. Li, M., Chen, Y., Kumar, N., Lal, C., Conti, M., Alazab, M.: Quantifying location privacy for navigation services in sustainable vehicular networks. IEEE Trans. Green Commun. Netw. **6**(3), 1267–1275 (2022). https://doi.org/10.1109/TGCN.2022.3144641
29. Liu, F., Zhao, X.: Disclosure Risk from Homogeneity Attack in Differentially Private Frequency Distribution (2021). https://doi.org/10.1109/TDSC.2022.3220592
30. Liu, F., Zhao, X.: Disclosure risk from homogeneity attack in differentially privately sanitized frequency distribution. IEEE Trans. Depend. Secure Comput., 1–12 (2022). https://doi.org/10.1109/TDSC.2022.3220592
31. Liu, J., Miao, Q., Fan, X., Wang, X., Lin, H., Huang, Y.: Mutual-supervised federated learning and blockchain-based IoT data sharing. Secur. Commun. Netw. **2022**, 1–8 (2022). https://doi.org/10.1155/2022/7003426
32. Ma, Y., Zhu, X., Hsu, J.: Data poisoning against differentially-private learners: attacks and defenses. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pp. 4732–4738. International Joint Conferences on Artificial Intelligence Organization, California (2019). https://doi.org/10.24963/ijcai.2019/657
33. Markelle Kelly Rachel Longjohn, K.N.: the UCI Machine Learning Repository (UCI). https://archive.ics.uci.edu/
34. Matthews, G.J., Harel, O.: Assessing the privacy of randomized vector-valued queries to a database using the area under the receiver operating characteristic curve. Health Services Outcomes Res. Methodol. **12**(2-3), 141–155 (2012). https://doi.org/10.1007/s10742-012-0093-y
35. McClure, D., Reiter, J.P.: Differential privacy and statistical disclosure risk measures: an investigation with binary synthetic data. Trans. Data Privacy **5**(3), 535–552 (2012). https://www.scopus.com/inward/record.uri?eid=2-s2.0-84871885755&partnerID=40&md5=e22f7e146ecad8931b9cddac0e7ebb15
36. Mendes, R., Vilela, J.P.: Privacy-preserving data mining: methods, metrics, and applications. IEEE Access **5**, 10562–10582 (2017)
37. Minna, K., Kärkkäinen, T., Hämäläinen, T.: Differential privacy: An Umbrella review. In: Tuomo, S., Kokkonen, T., Karjalainen, M. (eds.) Artificial Intelligence and Cybersecurity: Theory and Applications, pp. 167–183. Springer International Publishing, Cham (2023). https://doi.org/10.1007978-3-031-15030-2_8
38. Oguri, H.: A method of decreasing connectability of derived data, using local differential privacy. In: 2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1–8. IEEE, Piscataway (2019). https://doi.org/10.1109/ECAI46879.2019.9042011
39. Ohm, P.: Broken promises of privacy: Responding to the surprising failure of anonymization. 57 UCLA L. Rev., pp. 1701–1777 (2010)

40. Oksuz, A.C., Halimi, A., Ayday, E.: AUTOLYCUS: Exploiting Explainable AI (XAI) for Model Extraction Attacks against White-Box Models (2023)
41. Potter, B.: Microsoft SDL threat modelling tool. Netw. Secur. **2009**(1), 15–18 (2009). https://doi.org/10.1016/S1353-4858(09)70008-X. https://www.sciencedirect.com/science/article/pii/S135348580970008X
42. Primault, V., Boutet, A., Mokhtar, S.B., Brunie, L.: The long road to computational location privacy: a survey. IEEE Commun. Surv. Tutor. **21**(3), 2772–2793 (2019). https://doi.org/10.1109/COMST.2018.2873950
43. PubGenius Inc.: SciSpace. https://typeset.io/
44. Reilly, D., Fan, L.: A comparative evaluation of differentially private image obfuscation. In: 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), pp. 80–89. IEEE, Piscataway (2021). https://doi.org/10.1109/TPSISA52974.2021.00009
45. Rigaki, M., Garcia, S.: A Survey of Privacy Attacks in Machine Learning (2020). https://doi.org/10.1145/3624010
46. Salas, J., Torra, V., Megías, D.: Towards Measuring Fairness for Local Differential Privacy, pp. 19–34 (2023). https://doi.org/10.1007/978-3-031-25734-6_2
47. Shen, A., Francisco, L., Sen, S., Tewari, A.: Exploring the relationship between privacy and utility in mobile health: algorithm development and validation via simulations of federated learning, differential privacy, and external attacks. J. Med. Int. Res. **25**, e43664 (2023). https://doi.org/10.2196/43664
48. Shostack, A.: Threat Modeling: Designing for Security. Wiley, Hoboken (2014)
49. Sun, P.J.: Research on selection method of privacy parameter ϵ. Secur. Commun. Netw. **2020**, 1–12 (2020). https://doi.org/10.1155/2020/8845038
50. Sweeney, L.: Simple Demographics Often Identify People Uniquely (2000)
51. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncert. Fuzziness Knowl.-Based Syst. **10**(05), 557–570 (2002). https://doi.org/10.1142/S0218488502001648
52. Tankard, C.: What the GDPR means for businesses. Netw. Secur. **2016**(6), 5–8 (2016). https://doi.org/10.1016/S1353-4858(16)30056-3
53. Tran, H.Y., Hu, J.: Privacy-preserving big data analytics a comprehensive survey. J. Parall. Distrib. Comput. **134**, 207–218 (2019). https://doi.org/10.1016/j.jpdc.2019.08.007. https://www.sciencedirect.com/science/article/pii/S0743731519300589
54. United States: Health Insurance Portability and Accountability Act of 1996 (HIPAA). https://www.cdc.gov/phlp/publications/topic/hipaa.html
55. United States: The Family Educational Rights and Privacy Act (FERPA) (2021). https://www.govinfo.gov/app/details/USCODE-2021-title20/USCODE-2021-title20-chap31-subchapIII-part4-sec1232g/summary
56. Vokinger, K.N., Muehlematter, U.J.: Re-Identifikation von Gerichtsurteilen durch ≪ Linkage≫ von Daten (banken). Eine empirische Analyse anhand von Bundesgerichtsbeschwerden gegen (Preisfestsetzungs-) Verfügungen von Arzneimitteln. Zurich Open Repository and Archive, University of Zurich (2019)
57. Wagner, I., Eckhoff, D.: Technical privacy metrics. ACM Comput. Surv. **51**(3), 1–38 (2019). https://doi.org/10.1145/3168389
58. Wang, Y.R., Tsai, Y.C.: The protection of data sharing for privacy in financial vision. Appl. Sci. **12**(15), 7408 (2022). https://doi.org/10.3390/app12157408
59. Yan, H., Li, X., Li, H., Li, J., Sun, W., Li, F.: Monitoring-based differential privacy mechanism against query flooding-based model extraction attack. IEEE Trans. Depend. Secure Comput. **19**(4), 2680–2694 (2022). https://doi.org/10.1109/TDSC.2021.3069258
60. Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.Z., Li, H., Tan, Y.a.: Secure multi-party computation: theory, practice and applications. Inf. Sci. **476**, 357–372 (2019). https://doi.org/10.1016/j.ins.2018.10.024. https://www.sciencedirect.com/science/article/pii/S0020025518308338
61. Zhao, Y., Chen, J.: A survey on differential privacy for unstructured data content. ACM Comput. Surv. **54**(10s) (2022). https://doi.org/10.1145/3490237. https://doi-org.ezproxy.jyu.fi/10.1145/3490237

To Know What You Do Not Know: Challenges for Explainable AI for Security and Threat Intelligence

Sarah van Gerwen, Jorge Constantino, Ritten Roothaert, Brecht Weerheijm, Ben Wagner, Gregor Pavlin, Bram Klievink, Stefan Schlobach, Katja Tuma, and Fabio Massacci

1 Introduction

Threat intelligence (TI) builds upon many, sometimes unknown or unreliable sources and must operate under operational and legal constraints that cannot be interpreted by a single automated system. In a threat intelligence hybrid workflow (TIHW), human analysts and machines powered by artificial intelligence (AI) cooperate [12, 92]. Analysts routinely assemble findings derived from data generated by machines or assembled by other human analysts. These findings must often be assembled from data that analysts are not able to share or even have access to. Yet, TI must also be actionable to be useful for planning an intervention (such as apprehension of suspect of a cyberattack). Actionable information is relevant, timely, accurate, complete, and ingestible [79]. These properties are difficult to assert when the data itself cannot be accessed and/or when all the sources cannot be trusted.

S. van Gerwen (✉) · R. Roothaert · S. Schlobach · K. Tuma
Vrije Universiteit, Amsterdam, Netherlands
e-mail: s.a.m.van.gerwen@vu.nl

J. Constantino · B. Wagner
Delft University of Technology, Delft, Netherlands

B. Weerheijm · B. Klievink
Leiden University the Hague, Den Haag, Netherlands

G. Pavlin
Thales Research and Technology, Delft, Netherlands

F. Massacci
Vrije Universiteit, Amsterdam, Netherlands

University of Trento, Trento, Italy
e-mail: fabio.massacci@ieee.org

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_4

Consequently, at each step of TIHW, the analyst must revise this limited and uncertain information and recommend actions. For example, they must choose which explanation among many is more likely or ask for additional yet proportional investigations. Further down the line of the TIHW, part of the gathered intelligence may be used to determine the proportionality of interventions, an investigation conducted by oversight bodies [57]. For example, in the Netherlands, the Review Committee on the Intelligence and Security Services (CTIVD) investigates the lawfulness of conduct by the General Intelligence and Security Service (AIVD) and the Military Intelligence and Security Service (MIVD) and in 2022 reported that in case of cable interception (report no. 75), the duty of care had been insufficiently implemented.[1]

Yet, intelligence sharing remains challenging due to fear of negative publicity, legal constraints, quality issues, and prevalence of other uncertainties [81, 92] despite decades of research [104]. In addition, AI-powered solutions and human experts will always have biases [2, 54, 106] or imperfect models [78] which may further contribute to the overall uncertainty of the gathered intelligence and assembled findings.

The goal of this chapter is to discuss the emerging socio-technical implications and technical challenges in the formalization and quantification of uncertainty within threat intelligence. We will start with a description of the situation at hand (cf. Sect. 2), discussing of related work (cf. Sect. 3) in the threat intelligence environment. Thereafter, we will discuss socio-technical challenges within the legal, societal, and organizational field (cf. Sect. 4). Afterward, in Sect. 5, the technical challenges with regard to the formalization and empirical evaluation in the TIHW are presented. Finally, we close the chapter with an overview of the bigger picture.

2 The Problem of Threat Intelligence

Threat intelligence relies on analysts to bring information together and distill actionable intelligence from it [76, 109, 111]. Thus, what is distilled as actionable is a product of human decision-making. Expert judgment enables analysts to make decisions in real time in a different way than novices [75]. The intuitive form of expert judgment relies on the ability to make predictions about the environment and the possibilities to learn about the commonalities within the environment [53]. In other words, expert judgment is contingent on the situational awareness of the expert.

Figure 1 illustrates the process of decision-making in a threat intelligence scenario. The new information (green in Fig. 1) shows the meta-level information about the intelligence that could be useful for decision-making but is not always known (or available).

[1] https://english.ctivd.nl/investigations.

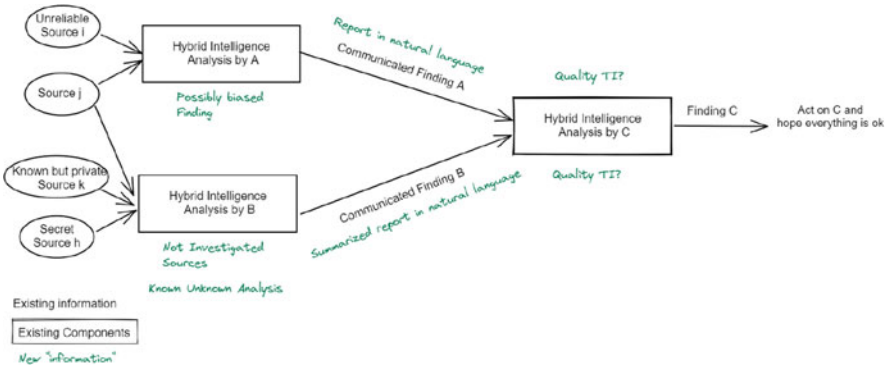


Fig. 1 The threat intelligence process and problem. *When A and B communicate their findings to C, it is hard for C to formally evaluate the uncertainty of the initial sources where A and B's findings are based upon*

Table 1 Overview of recently (2018–2023) researched biases and uncertainties in cybersecurity from the perspective of the analyst/defender

Uncertainty/bias	Description	Ref	Regarding
Overconfidence	Predictions are too certain or too uncertain given the actual performance	[33]	Managing a cyber-physical environment under threat.
Primacy bias	The first item in a series has the best recall	[38]	Attribution of cyber operations
Seizing and freezing	To combat cognitive dissonance, seizing shows a predisposition to information that confirms existing beliefs and freezing shows the refusal to adjust judgments to maintain beliefs	[39]	Attribution of degradative cyber operations
False sense of validation	When there is a perception of a human in the loop, action is undertaken on less and incomplete information	[107]	Use of AI in cyber conflicts
Information-pooling bias	In a team, information that is known by most members is more likely to be shared than information that is unique to an individual	[83]	Incident correlation in cybersecurity threat detection

Table 1 shows an overview of recently researched biases and reasoning with uncertainty (approximately 2018–2023) within the field of cybersecurity from the position of the analyst/defender. Furthermore, time pressures [46], height of stakes of decisions [50], secrecy [57], and a range of complexity are associated with intelligence problems [70]. Achieving situational awareness is difficult, because these characteristics make it hard to know, understand, and make predictions about the environment [53].

2.1 Adding Artificial Intelligence to the Equation

When using AI to obtain threat intelligence, we can speak of a double-edged sword. On the one hand, large datasets can be collected and processed to extract potentially useful information for analysts. On the other hand, this collection process can be expensive [89], invasive [13], and biased or unfair [78]. Furthermore, the sheer amount of information in combination with its dubious provenance and varying quality could result in increased confusion [92] or no actionable intelligence.

For example, in case studies [80], data on the entire Washington DC was collected, and AI-based techniques were used to predict criminal events with high accuracy. Yet, the predictions were not tactically actionable as the predicted hot spot areas amounted to the whole pedestrian area downtown. In contrast, systems with lower accuracy (e.g., 30%) but taking into account "awareness information" on uncertainty were considered more actionable by officers planning for Improvised Explosive Devices detection in Iraq [50].

2.2 Key Issues

Key issues in the field of threat intelligence are categorized as follows:

1. **Socio-technical context: legal and societal elements**—Providing threat intelligence in a genuine democratic society cannot focus solely on maximizing the quality of threat intelligence from a technical perspective. Threat intelligence needs to be accompanied by constitutional safeguards such as providing reliable and robust oversight systems and advancing privacy rights to ensure societal trust in security and intelligence operations. Engaging with this value multiplicity around threat intelligence is crucial to understanding the *legal and societal restrictions*, as well as the wider societal context. As noted by Laura Carlsen [14], "there can be no security without human rights."
2. **Socio-technical context: organizational elements**—Showing how risk-based decisions feed into professional practices and generate knowledge despite limited and constrained sharing [8] is not a trivial task. A structural change in work reflects an *organizational change.* In practice, organizational changes and learning only happen through its members [85], for the good or the bad in their social context [40]. Impact can only be achieved through concrete outcomes in the change of employees' daily work. Organizational learning is studied in environments where knowledge sharing is not heavily restricted [4]. The heavily restricted and sometimes misunderstood nature of threat intelligence sharing [104] results in different challenges viewed through this organizational lens.
3. **Uncertainty of information**—Sources and methods often have significant uncertainties and biases. Analysts are aware of these limitations, but uncertainty is yet to be captured and quantified within the threat intelligence workflow. The need arises for a formalization of uncertainty that can be both machine-readable and human-interpretable.

This undertaking is difficult due to the conditions of decision-making and information sharing in threat intelligence, a heavily restricted and possible deceptive environment [104]. Uncertainty can arise from multiple factors including potential observation errors, imprecision, communication errors, ambiguity, and unknown credibility of the sources. Quantifying and *formalizing uncertainties* can stem from qualitative concepts. These formalizations are distilled from a plethora of heterogeneous sources and need to be relevant in the decision-making process. This type of formalization results in specific challenges.

4. **Empirical evaluation**—The complex conditions of decision-making in threat intelligence workflows makes conducting *empirical validation* extremely challenging. Methodologies from existing literature (as further explained in Sect. 3.7) cannot be directly applied in the context of threat intelligence. For example, with respect to cybersecurity, existing experimental methodologies assume that the individual inputs to a method under scrutiny come from known sources, are complete and correct [59, 91, 97]. This conflicts with the reality of a threat intelligence workflow.

3 Related Work

In this section, we will discuss the related literature on information fusion, vocabularies for threat intelligence, uncertainty representation and reasoning, human judgment and communication of uncertainty, and experimental methodologies for threat intelligence.

3.1 Information Fusion

Information fusion systems extract actionable information from numerous sources [16]. The task of a threat intelligence analyst can be seen as an information fusion task. The first applications of information fusion combined simple sensory data in situations where the physical model was well-understood [67]. With the progression of AI, fusion systems are enabled to incorporate models that learn and adapt to complicated environments [67, 88]. These environments are characterized by data that comes from heterogeneous sources including (but not limited to) sensors, processes relying on learned models, and humans. One example is the DPIF platform where data and information sharing is supported during TIHW processes [100].

When information fusion systems increase in complexity (e.g., increase in amount of heterogeneous sources or overall scale), it also becomes increasingly difficult to make accurate inferences and support effective decision-making [67]. One key challenge within this difficulty is the role of uncertainty [16, 67, 100].

If uncertainty is not considered properly, fusion processes may deliver "underconfident, overconfident and/or incorrect results" [100].

3.2 Vocabularies for Threat Intelligence

Vocabularies exist for standardizing cyber threat intelligence. See Tounsi and Rais [92] for an overview of standards used for TI representation and sharing. Two of the most used vocabularies are STIX™ (Structured Threat Information eXpression) [20] and MITRE ATT&CK™ [22]. STIX™ is a structured language and serialization format that encompasses domain objects like attack pattern and campaign. It also includes relationship objects. MITRE ATT&CK™ is a vocabulary that is concerned with adversary tactics and real-world techniques. It encompasses a plethora of different techniques ranging from defense evasion to credential access. STIX™ and MITRE ATT&CK™ both provide detailed information that can be used to build knowledge graphs for cyber threat intelligence. However, what these vocabularies are missing is information about uncertainty.

3.3 Uncertainty Representation

The need for standardization of uncertainty representation has long been recognized [60]. However, creating a unified vocabulary applicable across multiple domains is difficult as the requirements for such a vocabulary may vary across these domains. As a result, attempts at creating a standardized vocabulary retain some level of domain-specificity (see Table 2).

Uncertainty within the context of information fusion is often discussed using the distinction between aleatoric and epistemic uncertainty. Aleatoric uncertainty arises from the variability in outcome due to randomness [45]. Thus, this type of uncertainty lies within the modeled environment [101]. Epistemic uncertainty refers to a lack of knowledge. This type of uncertainty refers to the epistemic state of an agent instead of random phenomenon [45]. Therefore, it lies outside of the modeled environment and might be mitigated by querying for additional information [101].

In De Villiers et al. [101], this distinction is used to help make uncertainty explicit during the information fusion process. Uncertainty can be factorized to include potential observation errors, imprecision of measurement, communication errors, ambiguity, unknown credibility of sources, and many more. Due to these different factors and categories, explicitness helps in reasoning about involved uncertainties throughout the fusion process. Whether these uncertainties have a one-to-one translation in the current domain remains to be seen.

The URREF ontology [23] is a work in progress and provides an overview of the potential sources of uncertainty. It provides the opportunity to explore the boundaries of the (information fusion) system that one is building. For a more

Table 2 An overview of previous attempts at creating a vocabulary regarding uncertainty representation from various domains

Domain	Relation to uncertainty	Example
Decision sciences	While the domain of decision sciences is incredibly diverse, it is rarely the case that conclusions can be drawn with full certainty; experiments are conducted in controlled environments, and models are developed based on imperfect data. Therefore, proper communication of findings requires reporting the uncertainties associated with those findings	[32, 87]
Earth system science	Earth systems highlight a different challenge when representing uncertainty: spatiotemporal scaling. Atmospheric models can be reasonably accurate when considering a daily global model output but may fail to provide any usable insights on lower scales. Alternatively, small-scale population estimates might not generalize to large-scale ecosystems. This, along with the stochastic nature of ecological processes, measurement error, and human judgment, is an important source of uncertainty within earth systems	[86, 110]
Database management	Data and uncertainty are closely intertwined. Uncertainty would not exist without data, and most, if not all, forms of data is to some degree uncertain. In the context of database management, this means that protocols are needed on how to combine the uncertainty information when merging data sets. These protocols not only depend on the type of uncertainty but also on the domains from which the data originates	[61]
Information fusion	When developing an information fusion system, the uncertainties associated with the fused information determine which fusion method can be applied. Therefore, a careful evaluation of those uncertainties is needed at the start of the development process	[23]

complete account of representing uncertainty in decision-making, we refer the interested reader to the survey of Keith and Ahner [56].

3.4 Reporting Uncertainty in Threat Intelligence

The most basic and most important task of actionable threat intelligence is reporting this intelligence to decision-makers. Reporting uncertainty successfully with respect to threat intelligence has been deemed an important area of study [65]. This is the case, because there is an effect of the way uncertainty is represented on decision-making [29].

Natural language in the form of linguistic categories (e.g., “Likely” or “Probable”) is often used to represent uncertainty within threat intelligence. An example is the Admiralty Code used in several intelligence organizations [46]. Research has shown [65] that this way of conveying uncertainty is often ineffective due

to differences in the way these categories are interpreted by humans, even when definitions are set.

A possible alternative would be using numerical representations [27]. One of the reasons why numerical representations have yet to be implemented is the fear of more risky decision-making by analysts. Indeed, Friedman et al. [35] found that less experienced analysts were overconfident in their decision-making when using numerical representations of uncertainty. However, the same study also found that, generally, numerical representations were actually associated with less risky decision-making and more accurate predictions. These findings suggest that experience might overcome this overestimation [35].

Another possibility is that linguistic categories may contain more than just information of probability [18]. Collins and Mandel [18] proposed that linguistic categories could be used during deliberate argumentation, while numeric categories were most suitable in situations where clear probability estimations were required. A combination of the two formats was also investigated, although no significant difference was found in performance of numerical representation and the combined representation [66]. However, both formats were more effective than just linguistic categories when it came to probability estimation.

Graphical and visual representations for uncertainty representation have also been researched [77]. For example, in a dynamic decision-making missile-defense game, support was found for using graphical representations to increase efficiency even in combinations with numerical representations [7]. For a more complete overview of visual representations, see [77]. Within the context of cyber threat intelligence in a national security environment, as far as the authors are aware, it is not clear how these different representations feed in to professional practices.

3.5 Human Judgment and Bias Under Uncertainty

Decision-making in threat intelligence includes decisions about security risks which are made in face of uncertainty [6], leaving space for subjective and possibly biased judgment [49]. This type of uncertainty is a key element in socio-technical systems.

Judgment under uncertainty has been the object of study for decades [34, 54, 71]. In the past, the focus has been on looking for reasoning shortcuts (heuristics) and bias within human decision-making. This research is built upon by theories of Bayesian inference (the framing of a problem leads humans to view new information in accordance with prior beliefs) [108] and in opposition and digitization (reasoning with uncertain information leads humans to favor the most certain information and ignoring other information [52]). In addition, the dual processing theory expands the existing framework (humans use two systems in the decision-making process, one for quick automatic judgments and one for deliberative, slow, and complex calculations) [108]. For an overview of biases and debiasing techniques within the general field of decision and risk analysis, see Montibeller and Von Winterfeldt [71], and cybersecurity, see Johnson et al. [51].

However, even though the previously mentioned research on known biases (see Table 1), the influence of AI, and team-level heuristics exists, research in reasoning with bias and uncertainty is sparse and scattered over many categories in cybersecurity. Furthermore, the population of intelligence analysts differs from other populations. Especially in the world of cyber professionals, due to heterogeneity in work roles and skill sets, findings are population specific [68]. Whether the same biases and reasoning techniques are relevant within the current field remain to be investigated.

3.6 Existing Approaches to Help Elicit Expert Knowledge in Threat Intelligence

Analysts are highly trained experts that use intuitive decision-making to deal with situations [75]. Okoli et al. [75] put forward that the consensus in literature is that experts are able to make quicker and often better decisions because they are able to use their existing knowledge to assess the situation at hand with the usage of schema. These schema (i.e., strong memory networks) allow experts to have a perceptual advantage even when events unfold in real time [75].

Expert knowledge elicitation techniques are techniques that try to improve the quality of expert judgment with respect to debiasing and reasoning under uncertainty [28]. To that end, structured analytic techniques (SATs) have been previously used in the domain of threat intelligence. SATs are techniques that systematically and transparently aim to externalize internal thought processes [106]. SATs are often not well researched [65], and the few existing studies have shown mixed results [17, 106].

In cases where the SATs have been implemented, an important challenge is to either update these techniques to handle cyber threats and measure their efficacy [17] or develop new ways to make human reasoning transparent and enable explainability. In the field of general decision and risk analysis, see Dias, Morton, and Quigley for an overview [28] of expert knowledge elicitation techniques.

Another approach to make judgments less biased is the use of coherentization and aggregation of judgments [55]. Here, multiple numerical judgments from different analysts are first made coherent with respect to specific statistical assumptions (e.g., probabilities must add up to 1) and, thereafter, aggregated into one prediction. However, this approach has not been researched with respect to realistic cyber threat intelligence scenarios.

In intelligence analysis, meta-information is already used to aid decisions and judgments under uncertainty. The standard NATO Standardization Agreement (STANAG) 2511 captures two qualitative categories [65], although their exact implementation is not uniform between different intelligence agencies [46]. In general, source reliability can be seen as a confidence level based on historical performance. Information credibility captures the extent to which a new piece of

information is consistent with the current reporting [46]. However, these factors are often not enough to provide the necessary uncertainty information. For example, source reliability can vary tremendously in different situations. The current categories do not provide a way to make this distinction explicit [46].

3.7 Existing Experimental Procedures and Methods

Existing approaches assume that the individual inputs to a method under scrutiny come from known sources, are complete, and are correct [59, 91, 97]. Representations of security and malicious threats (e.g., attack and defense trees, data flow diagrams, petri-nets, etc. [95]) are compared by either observing the quality of the representation (compared to a baseline model) [31] or by observing some measure of the analysis output (e.g., the precision of the identified security threats [97] or complexity of generating all attack paths [44]).

Due to deception, the variability in sources, and incompleteness (automatically generated), cyber threat intelligence can only be actionable if it takes the quality of the information into account. To underline this point, Ranade et al. [84] generated fake cyber threat intelligence and observed that experts would consider both the deceptive TI and the authentic TI as equally true. Without an explicit discussion on the quality of TI, these effects remain unnoticed.

Decision support systems can also be used in cybersecurity to assist analysts in their job [11, 37]. For instance, decision support systems can be used to aid optimizing cyber forensic investigations [73], cybersecurity threat and incident management [98], and the manual assessments of proportionality in military cyber operations [64]. The evaluation of decision support systems mainly focused on conducting user studies and evaluating transparency (i.e., explain how the system works) and trust (increase user confidence in the system) of the decision support system [74].

Methods in explainable AI (xAI) [43] are certainly interesting to investigate for the purpose of evaluating persuasiveness. In a recent survey on the evaluation of xAI systems, Nauta et al. [72] found that only one fifth of the analyzed papers evaluated their findings with users. In addition, Dalvi et al. [26] claim that within cyber threat intelligence, multiple xAI implementations to help understand AI algorithms do not actually agree with each other on their explanation. This is no surprise because there is no standard correct or best explanation when it comes to measuring explanations in many scenarios [74].

4 Socio-technical Challenges

In this section, challenges with respect to the socio-technical context of a TIHW are discussed. In a multidisciplinary approach, legal, societal, and organizational

matters are discussed. For each of the identified challenges, ideas are proposed to handle them.

4.1 Identification and Operationalization of Key Societal and Legal Elements

Governments and their agents are expected to follow the law, particularly in sensitive constitutionally protected matters such as privacy, while ensuring that other key rights are also safeguarded [48]. In the larger governance context, political decision-makers need to take these challenges into account to monitor, oversee, and evaluate responsible intelligence and security agencies accordingly. For instance, passing certain laws to facilitate the work of security and intelligence services may support ethical decision-making or would nudge security and intelligence operations to fall into a web of unethical practices [58]. These considerations contribute to strengthening citizens' trust toward security and intelligence operations, particularly where the provision of data or information is needed [24].

At the same time, intelligence and security agencies need to ensure that their practices remain within the boundaries of the law and the core of public service integrity [58]. In this environment, agencies can implement innovations, and government can realign their model in a way that supports confidence in public service by putting the well-being of constituents first above all [21].

The identification and implementation of key elements is hard. For example, transparency has the ability to obscure (e.g., showing so much data to distract from the central information) [3], can encourage "seeing" over understanding (i.e., being able to look inside a system is not enough, one also needs to be able to interact with them in a broader social context to have an actual understanding) [3], and can be used to create an atmosphere of transparency in the public eye instead of fostering accountability within an organization [1].

Socio-technical challenge Sect. 4.1: It is unclear what and how legal and societal constraints can be implemented in a TIHW
The identification and implementation of key legal and societal elements is necessary to ensure that intelligence and security agencies' practices are lawful and enable trust in citizens. Currently, it is not known how legal and societal constraints need to be incorporated in professional hybrid threat intelligence practices.

We are in the process of identifying key societal and legal elements of the TIHW. The workflow needs to follow the standard principle of necessity established under international human rights law: establishing an objective goal deemed to be

necessary in order to protect a legitimate interest (e.g., specific target affecting national security) [57].

Having established necessity, we can then address the proportionality test, which requires the workflow to establish a justification balancing the methods to be utilized against the intended goal. For instance, a TIHW should conduct impact assessments to determine proportionality and assess whether the workflow causes a chilling effect on citizens. In addressing a subsidiary analysis, the workflow will be established in an environment where its framework is regulated by transparency.

It is first necessary to develop a framework based on international best practices of the socio-technical conditions and constraints for risk-based decisions in intelligence communities such as necessity and proportionality. How do comparable threat intelligence procedures take place in other countries and governmental contexts?

Second, an in-depth juridical analysis of relevant laws and regulations should be conducted to inform the design of the TIHW. This can also help understand how international best practices can be integrated into existing workflows and how these could be operationalized in practice.

Idea Sect. 4.1: Develop a framework based on international best practices and relevant laws and regulations
Conducting research in relevant laws and regulations and international best practices can show how key legal and social elements can be integrated into existing workflows and be operationalized in practice.

4.2 Future Proofing Data Protection and Human Right Safeguards

Technology changes rapidly. For example, big tech companies currently try to construct methods for identifying deepfakes while internet enthusiasts and state-sponsored disinformation campaigns keep finding new systems to fool these detectors [25]. This arms race leaves regulations to be outdated.

Even in a perfect world where all limitations would have concrete definitions that were agreed upon, the issue remains that not all limitations can be satisfied at all times. What is perceived by some as an easy solution, for example, mass surveillance of communications, may be the most harmful approach from a privacy or digital rights perspective. Furthermore, these trade-offs also encompass an economical aspect. Although security (e.g., finding threats) is the goal, analyst's time is often considered more important in the decision-making process, since this time is expensive and scarce [12]. To address these limitations in a way that has a better chance of working in practice over a substantial amount of time remains a challenge.

Socio-technical challenge Sect. 4.2: Technology changes rapidly, and limitations cannot be satisfied at the same time
Limitations of the system have to be addressed in a way that has a better chance of working in practice over a substantial amount of time.

Future proofing efficacy and safeguards should include both considering incoming international legal frameworks such as Council of Europe Convention 108+ [30] and responding to existing societal challenges such as the engagement of commercial Open Source Intelligence (OSINT) which might be caught by the future enactment of the European Artificial Intelligence Act [19].

Idea Sect. 4.2: Incoming international legal frameworks and existing societal challenges should be addressed in a TIHW
Future proofing data protection and human rights safeguards should be taken into account to make the workflow not only relevant now but also robust in the future.

4.3 Studying Organizations with Restricted Information Sharing

The amount of information expands rapidly, and no one institution or vendor can hold it all. Bouwman [12] found that, when looking at two commercial providers of threat intelligence, there was minimal overlap in the indicator feeds, even in cases of identical threat actors. Next to processes between organizations, information sharing is also influenced by internal organizational processes. Domains (e.g., military, national security) within threat intelligence work in compartments to limit information flow [58].

Decision-makers, analysts, and field personnel do not have access to the same information. In environments where information sharing is restricted due to need-to-know policies, the sharing of healthy behavior toward regard for human dignity and ethics may also be difficult. Due to compartmentalization, sometimes best practices are difficult to transfer to other compartments. Even when there is a transfer of colleagues between compartments, there is a risk of mixing between colleagues upholding "positive" culture and colleagues caught in "negative" work culture [58]. Instead of a chain of restricted information sharing, decision-makers and analysts have the opportunity to act as autonomous agents [99].

Furthermore, individual motives play in a role in the information sharing as well. Found information can, depending on the significance of said information, result in increased "status" once shared with a superior [102]. The higher the position of

the superior in the organizational hierarchy, the greater the potential "status" gain. Sharing information with peers or superiors lower in the organizational hierarchy could result in them taking credit, leaving the original finder with a diminished "status" gain.

Socio-technical challenge Sect. 4.3: Analyzing/improving information sharing in a restricted environment is difficult
External organizational processes, internal organizational processes, and psychological factors influence the degree of information sharing. It is unclear how these elements feed in to organizational practices within a restricted environment.

An integrative view of the three levels of organizational change is necessary to get a better understanding of the impact of change during the implementation of a system. These levels consist of an operational level (learning of the actual professionals themselves), a tactical level (learning from a management perspective), and a strategical level (learning in the perspective of organizational compliance and regulations). In addition, this integration helps in pinpointing effects that would otherwise stay unseen. This view calls for feedback loops both within the hybrid intelligence pipeline and across these levels, within and between organizations.

A potential solution and opportunity for empirical work could be the implementation of an acceptable (by the organization and the employees) variation on the nice-to-know code among different compartments; this may facilitate the transfer and fostering of values and international principles such as proportionality.

Idea Sect. 4.3: Integrating the operational, tactical, and strategic levels of organizational change with a clear distinction of processes between organizations and those within organizations
Taking into consideration the internal processes of an organization (individual motives, team dynamics, and organizational aversion to change), and the external processes (between organizations, in a broader societal context), using empirical evaluation, can help pinpointing unforeseen effects.

5 Technical and Experimental Challenges

In this section, technical and experimental challenges of formalizing uncertainty within a TIHW are discussed. For each of the identified challenges, ideas are proposed to handle them.

5.1 Representing Uncertainty Stemming from Systems, Humans, and Situations

As discussed in Sect. 3.4, uncertainty within threat intelligence workflows is mainly conveyed in natural language between people. Standardized natural language formats are used in certain domains. For example, within the defense domain, the standard NATO Standardization Agreement (STANAG) 2511 incorporates linguistic labels to communicate source reliability and information credibility [65]. However, such uncertainty representation is typically not suited for machine-based processing. Since these uncertainties often stem from qualitative concepts, it can be challenging to translate them into representations that quantify the uncertainty for the machine-based processing and vice versa. Furthermore, the threat intelligence workflow is hybrid, meaning that the uncertainties themselves will not only stem from abstractions and errors in systems and data but also from the process of human decision-making. In addition, to provide a basis for accountability in the larger societal context, uncertainty information should be available along the chain of communication.

Technical challenge Sect. 5.1: Representing and tracking uncertainty for actors in TIHW is complicated due to qualitative sources of uncertainty
Uncertainty will not only arise from abstractions and errors in systems and data but also from the process of human decision-making. The uncertainty information should be available along the chain of communication. Representing uncertainty stemming from qualitative concepts is challenging.

Uncertainty can be factorized by a plethora of elements. Due to these different factors and categories, explicitness helps in reasoning about involved uncertainties throughout the process.

A possible extension of the URREF ontology [23], introduced in Sect. 3.3, could serve as a basis to start representing uncertainty within the threat intelligence workflow. It provides the opportunity to explore the boundaries of the (information fusion) system that one is building. It also has the expressiveness to incorporate the current NATO-STANAG 2511 standard [9]. It should be considered a checklist, forcing any information system developer to thoroughly analyze the information fusion pipeline and make adjustments where necessary.

For uncertainty to be used not only between two agents who are in immediate connection to one another but also along the chain of communication, it is necessary that uncertainty provenance is tracked. A second proposed framework, handling provenance tracking, is the PROV data model [41]. This model can be used to structure the information in a knowledge graph (KG), making a distinction between entities (things that contain information), activities (the process that produced the information), and agents (persons/software/machines responsible for the taken

actions). If applied within a TIHW, it allows for the construction of a provenance trail of information, providing insights into origin of the information.

Idea Sect. 5.1: Making uncertainty explicit by expanding and combining existing ontologies
Uncertainty can be factorized by a plethora of elements. Due to these different factors and categories, explicitness helps in reasoning about involved uncertainties throughout the fusion process. A possible extension of the URREF ontology [23] in combination with the PROV ontology [41] could serve as a basis of representing these uncertainties.

5.2 Formal Reasoning with Uncertainty

When it comes to providing explanations that agree with human understanding, uncertainty representation is not enough. A specific type of formal uncertainty reasoning that can reflect abductive inference is necessary [69].

With respect to formal reasoning, two directions can be distinguished. These directions are forward and backward reasoning; see Fig. 2. Backward reasoning, in the current context, is about recreating trails and possibly gathering more information to demonstrate proof of the proportionality and subsidiarity of actions for each TIHW component and for the entire TIHW. Forward reasoning, in the current context, could be used in building an actionable strategy with minimal uncertainty. The challenge at hand is that there are no such reasoning tasks that minimize the uncertainty on process level for threat intelligence and also, with respect to proportionality and other legal and societal constraints, in a domain agnostic way.

Fig. 2 Forward and backward reasoning. *Forward and Backward Reasoning: Forward reasoning can be thought of as a form of what-if reasoning where the reasoning starts from the information and moves forward. Backward reasoning can be thought of as an evaluation where the reasoning starts from the decision and moves backward. The green color represents the respective reasoning paths*

Technical challenge Sect. 5.2: Formalizing reasoning tasks to minimize uncertainty is challenging
How to formalize reasoning tasks that minimize the uncertainty on process level, e.g., by backward reasoning ("what-if") or forward reasoning ("why")?

A third framework is needed, one that could unify the uncertainty overview with formalized reasoning about uncertainties. Which to choose is not a trivial choice. This depends on the complexity of reasoning capabilities, and the richness of the uncertainty overview. The combination of uncertainty representation, provenance tracking, and reasoning/causal inference is necessary.

Past approaches in uncertainty reasoning use fuzzy logic [10], epistemic logic [5], Markov network/processes [90], probabilistic logic [42], Bayesian networks [15], and Dempster-Shafer theory [112]. However, automated reasoning over uncertainty is very complex. The choice of reasoning method is dependent on the way the uncertainty is represented (e.g., in a qualitative format [105]). The combination of the URREF ontology [23] and PROV data model [41] with the intent to reason about uncertainty is particularly difficult. The foundations of both the URREF ontology and PROV data model are based on Boolean statements, either something is true or it is not. When dealing with uncertainties, the assigned value lies most often somewhere in the gray area in between.

Idea Sect. 5.2: Combine uncertainty representations with a provenance framework and reasoning/causal inference
To tackle provenance tracking and enable forward and backward reasoning within the workflow, the URREF ontology can be combined with the PROV framework [41] and integrated with a third framework (uncertainty reasoner) to unify the uncertainty overview with formalized reasoning about uncertainties.

5.3 *Experimental Methods Aware of Uncertainty*

Designing methods to evaluate the correctness of a threat intelligence decisions regarding a course of action or event likelihood is not easy, because the lack of ground truth is persistent in threat intelligence. In controlled experimentation, one possible solution is to curate the ground truth manually [93, 96], but this is not always feasible when analyzing large number of threat intelligence sources [62]. In previous work, there exists an optimal choice [55], preferences were measured

[47, 63], or expert judgment was used for validation [64]. These measurements were sometimes in a qualitative format, e.g., interviews [82, 103], and in other cases, they were quantitative, e.g., optimal likelihood estimations [55]. However, it is not clear how incomplete and uncertain threat intelligence information should be treated in the ground truth.

When it comes to expert judgment, the situation becomes complex. On the one hand, analysts are highly trained individuals in high-risk decision-making [75]. The combination of training and experience often leads to "intuitive" decision-making. This behavior is rarely seen in non-experts [75]. On the other hand, decisions about security risks may be affected by biased judgment [6, 49]. Uncertainty and bias are key elements of each socio-technical system. Minimization is not the ultimate goal. However, existing experimental methods lack protocols that can effectively and systematically measure human bias in threat intelligence decision-making [94].

Technical challenge Sect. 5.3: Existing empirical protocols for THIW validation have to be adapted to incorporate the human factor
What existing empirical protocols and measures can be adapted to quantify measures of uncertainties including human bias in TIHW?

The property of the exchanged information in the TIHW is that from an analyst's (or study participant's) point of view, the information may (or may not) be aggregated, incomplete, inaccurate, unreliable, and/or censored. And yet, a sound and convincing explanation (with at least partial traces in the model) for a minimal intervention (i.e., proportionality) must be possible. The current landscape of empirical methods does not cater for investigating such aspects of decision support systems. Important is to measure the human effects. Qualitatively and quantitatively evaluating the entire intelligence pipeline thus calls for novel protocols, measures, and controls to be developed.

See Table 3 for an overview of methods used in recent (2018–2023) research on bias and uncertainties in cybersecurity from the perspective of the analyst/defender (for background information on the research in question, see Table 1). Internal validation of surveys, questions, and other methods were rare. External validation of these methods was most often checked with a group of experts or participants [33, 38, 39]. These findings suggest that there is a need for more internal and external validation methods.

Validation in isolation is not insightful enough. A validation methodology has to be adapted to effectively assess heterogeneous systems with both AI, human, and unknown components.

Table 3 Overview of methods used in recent (2018–2023) research on bias and uncertainties in cybersecurity from the perspective of the analyst/defender

Uncertainty/bias	Ref	Type of study	Measures
Overconfidence	[33]	Cyber game	Argumentation and self-confidence via coded transcripts of verbal discussion
Primacy bias	[38]	Vignette	Attribution via survey on confidence levels
Seizing and Freezing	[39]	Vignette	Attribution via survey on confidence levels and coded justification
False sense of validation	[107]	Vignette	Machine preference via survey on confidence levels and selecting decisions
Information-pooling bias	[83]	Synthetic task environment (i.e. less focus on realism and more on the cognitive task at hand) experiment	Team collaboration and information pooling were measured via coded transcripts of the verbal discussions

Idea Sect. 5.3: Validating effectiveness of a human-based decision-making process (such as TIHW) calls out for human-in-the-loop experimental protocols
To this aim, new experimental protocols must be specifically designed to measure human effects. For instance, similar protocols outlined in [94] could be retrofitted to the domain of threat intelligence.

5.4 Computing with Objects of Evaluation to Measure Their Quality May Not Be Possible

Since direct computation over unknown (or uncertain) values is not possible, the evaluation should take as input meta-information rather than the object of evaluation. So the key question is not whether say an AI image recognition tool works with 80 or 90% of accuracy, but rather, which representation of such uncertainty is actionable for the user. However, methodologies for threat modeling and analysis and the protocols used for their evaluation require the user to specify the sources of security relevant components and the locations where such information is not allowed to flow. Therefore, existing methodologies [31, 44, 93, 95] cannot be directly carried over to evaluate the appropriateness of alternative suggestions by the TIHW, such as an alternative plan of intervention in the presence of a terrorist threat by requesting input from a new source.

Technical challenge Sect. 5.4: Measuring meta-information about objects of evaluation is necessary instead of measuring the object level
How can meta-information about objects of evaluation be measured and under what conditions are these measurements valid?

Since computation with the object of evaluation itself is not always possible, we need to make use of the meta-information that is available (e.g., timestamp, type of device, etc.) to define and compute new measures of quality. As put forward in Zibak, Sauerwein, and Simpson, data quality in threat intelligence has not been properly empirically investigated [113]. To achieve this, the first step is to investigate what type of meta-information is available from the field.

Confounding factors should be balanced within these measurements. For example, a THIW relies on AI modules, which can be symbolic modules explicitly taking uncertainty into account, or sub-symbolic modules (ML-like). For the latter, several studies exist on estimating and propagating uncertainty on the output of, e.g., deep learning models (see, e.g., the popular dropout method [36]), but there is no protocol to propagate the effect of hybrid errors of the next TIHW component.

Idea Sect. 5.4: Validate new measures to quantify meta-information about objects of evaluation
Since computation with the object of evaluation itself is not always possible, we need to make use of the meta-information that is available (e.g., timestamp, type of device, etc.) to define and compute new measures of quality. To achieve this, the first step is to investigate what type of meta-information is available from the field.

6 The Bigger Picture

In this chapter, we discussed the interplay between complex conditions and trade-offs between security and legal, societal, and organizational restrictions that make decision-making under uncertainty a challenging endeavor. Table 4 shows an overview of the illustrated challenges.

In the quest to achieve efficiency and effectiveness in threat intelligence, security and intelligence agencies are implementing AI-powered solutions to find actionable information to aid them in decision-making during uncertainty. However, one must remember that these AI tools to help deal with uncertainty in threat intelligence can end up being a double-edged sword. The development of a threat intelligence hybrid

Table 4 Overview of socio-technical, technical, and experimental challenges discussed in this chapter

Category	Section	Challenge	Idea
Socio-technical	Section 4.1	It is unclear what and how legal and societal constraints can be implemented in a TIHW	Develop a framework based on international best practices and relevant laws and regulations
	Section 4.2	Technology changes rapidly and constraints can not be satisfied at the same time	Incoming international legal frameworks and existing societal challenges should be addressed in a TIHW
	Section 4.3	Analyzing/improving information sharing in a restricted environment is difficult	Integrating the operational, tactical and strategic levels of organizational change with a clear distinction of processes between organizations and those within organizations
Technical	Section 5.1	Representing and tracking uncertainty for actors in TIHW is complicated due to qualitative sources of uncertainty	Making uncertainty explicit by expanding and combining existing ontologies
	Section 5.2	Formalizing reasoning tasks to minimize uncertainty is challenging	Combine uncertainty representations with a provenance framework and reasoning/causal inference
Experimental	Section 5.3	Existing empirical protocols for THIW validation have to be adapted to incorporate the human factor	Validating effectiveness of a human-based decision making process (such as TIHW) calls out for human-in-the-loop experimental protocols
	Section 5.4	Measuring meta-information about objects of evaluation is necessary instead of measuring the object level	Validate new measures to quantify meta-information about objects of evaluation

workflow (TIHW) is not an exception. Uncertainty will likely arise due to communication errors, ambiguity, and unknown credibility of the sources/provenance.

Challenges arise when creating a robust system that advances the embedding of regard for citizens' fundamental rights and responding to efficiency and support of user autonomy to enable intelligence agencies to arrive at the best possible decisions. Achieving this fine point is essential in a democratic society, because it develops societal trust in security and intelligence operations.

Despite developing a system that meets the requirements mentioned in this paper, we also need a path forward in security and intelligence operations to transfer this knowledge within their agencies or organizations. We recommend applying a lens based on international standardized legal principles, such as proportionality and necessity, during human AI interactions or evaluations in the absence of ground truth. Thus, the relationship between developing an AI system and having regard for societal and legal matters are not far from each other.

Uncertainty in a TIHW stems from quantitative as well as qualitative sources. This makes the formalization of uncertainty hard. In addition, uncertainty representation has to be machine-readable, as well as human understandable. Therefore, uncertainty representation should enable reasoning according to abductive inference. Representation and reasoning methods for uncertainty that capture these conditions have not been constructed with respect to threat intelligence.

AI augmented socio-technical systems for threat intelligence must respond to relevance, timeliness, accuracy, completeness, and ingestibility. A TIHW evaluation will require investigating the persuasiveness (e.g., to the oversight body), efficiency (helps analysts make decisions faster), and debugging (helps analysts identify when something is wrong and explore "what-if" scenarios) of the explanations, for which appropriate measures are to this day less explored.

The validation methodology for a TIHW has to holistically incorporate AI, human, and unknown components. In addition, confound-aware methods that measure the meta-level instead of the object-level of a TIHW are necessary. Validation methodologies within threat intelligence that satisfy these requirements have not been thoroughly investigated.

We hope to stimulate discussion and further research in the community by illustrating these challenges and possible ways to answer them.

Acknowledgments We are thankful to Sarah Giest and Iris Cohen for their valuable feedback. This work was funded by the *Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)* under the HEWSTI Project under grant no. 14261.

Contributions SVG (Sects. 2, 3, 4, 5, 6, Figs. 1, 2, Tables 1, 3, 4), JC (Sects. 4 and 6), RR (Sects. 5.1, 5.2, Fig. 2, Table 2), B Weerheijm (Sect. 4.3), B Wagner (Sect. 4), GP (Sects. 5.1 and 5.2), BK (Sect. 4), SS (Sects. 5.1 and 5.2), KT (Sects. 1, 2, 5.3, 5.4, Fig. 1), and FM (Sects. 1, 2, 6, Fig. 1) have conceived the presented ideas and contributed to their corresponding sections in the manuscript.

References

1. Albu, O.B., Flyverbom, M.: Organizational transparency: conceptualizations, conditions, and consequences. Business Soc. **58**(2), 268–297 (2019). https://doi.org/10.1177/0007650316659851
2. Alexander, P.: Exploring bias and accountability in military artificial intelligence. 7 LSE Law Review, pp. 396–405 (2022)
3. Ananny, M., Crawford, K.: Seeing without knowing: limitations of the transparency ideal and its application to algorithmic accountability. New Media Soc. **20**(3), 973–989 (2018). https://doi.org/10.1177/1461444816676645
4. Argote, L., Miron-Spektor, E.: Organizational learning: from experience to knowledge. Organiz. Sci. **22**(5), 1123–1137 (2011). https://doi.org/10.1287/orsc.1100.0621
5. Banerjee, M., Dubois, D.: A simple logic for reasoning about incomplete knowledge. Int. J. Approx. Reason. **55**(2), 639–653 (2014). https://doi.org/10.1016/j.ijar.2013.11.003. https://www.sciencedirect.com/science/article/pii/S0888613X13002478
6. Bier, V.: The role of decision analysis in risk analysis: a retrospective. Risk Analy. **40**(S1), 2207–2217 (2020)
7. Bisantz, A.M., Cao, D., Jenkins, M., Pennathur, P.R., Farry, M., Roth, E., Potter, S.S., Pfautz, J.: Comparing uncertainty visualizations for a dynamic decision-making task. J. Cogn. Eng. Decis. Making **5**(3), 277–293 (2011). https://doi.org/10.1177/1555343411415793
8. Blagden, D.: The flawed promise of national security risk assessment: nine lessons from the british approach. Intell. Nat. Secur. **33**, 716–736 (2018)
9. Blasch, E., Laskey, K., Jousselme, A., Dragos, V., Costa, P., Dezert, J.: URREF reliability versus credibility in information fusion (stanag 2511). In: Proceedings of the 16th International Conference on Information Fusion, FUSION 2013 (2013)
10. Bobillo, F., Straccia, U.: Fuzzydl: an expressive fuzzy description logic reasoner. In: 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), pp. 923–930 (2008). https://doi.org/10.1109/FUZZY.2008.4630480
11. Bohanec, M.: Decision support. In: Mladenić, D., Lavrač, N., Bohanec, M., Moyle, S. (eds.) Data Mining and Decision Support, vol. 745. The Springer International Series in Engineering and Computer Science. Springer, Berlin (2003). https://doi.org/10.1007/978-1-4615-0286-9_3
12. Bouwman, X., Griffioen, H., Egbers, J., Doerr, C., Klievink, B., van Eeten, M.: A different cup of TI? The added value of commercial threat intelligence. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 433–450 (2020)
13. Brown, I., Korff, D.: Terrorism and the proportionality of internet surveillance. Eur. J. Criminol. **6**, 119–134 (2009)
14. Carlsen, L.: Mexico's false dilemma: human rights or security. Nw. J. Hum. Rts **10**(3), 145–135 (2012)
15. Carvalho, R.N., Laskey, K.B., Costa, P.C.G.: PR-OWL – a language for defining probabilistic ontologies. Int. J. Approx. Reason. **91**, 56–79 (2017). https://doi.org/10.1016/j.ijar.2017.08.011. https://www.sciencedirect.com/science/article/pii/S0888613X17301044
16. Catano, V., Gauger, J.: Information fusion: Intelligence centers and intelligence analysis. In: Goldenberg, I., Soeters, J., Dean, W.H. (eds.) Information Sharing in Military Operations, pp. 17–34. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-42819-2_2
17. Claver, A., van de Meeberg, H.M.: Devil's advocacy within dutch military intelligence (2008–2020): an effective instrument for quality assurance? Intell. Nat. Secur. **36**(6), 849–862 (2021). https://doi.org/10.1080/02684527.2021.1946951
18. Collins, R.N., Mandel, D.R.: Cultivating credibility with probability words and numbers. Judg. Decis. Making **14**(6), 683–695 (2019). https://doi.org/10.1017/S1930297500005404
19. Commission, E.: Regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts (2021). COM(2021), 206 final, 2021/0106 (COD)

20. Committee, C.T.I.T.: Introduction to stix. https://oasis-open.github.io/cti-documentation/stix/intro.html (2023). Accessed 15 Jun 2023
21. Constantino, J.: Exploring article 14 of the eu ai proposal: Human in the loop challenges when overseeing high-risk ai systems in public service organisations. Amsterdam Law Forum **14**(3), 17 (2022)
22. Corporation, T.M.: Mitre att&ck. https://attack.mitre.org/ (2023). Accessed 15 Jun 2023
23. Costa, P., Jousselme, A.L., Laskey, K.B., Blasch, E., Dragos, V., Ziegler, J., de Villiers, P., Pavlin, G.: Urref: uncertainty representation and reasoning evaluation framework for information fusion. J. Adv. Inf. Fusion **13**(2), 137–157 (2018)
24. Court, T.H.D.: Njcm et al. v. the dutch state (2020). https://uitspraken.rechtspraak.nl/#!/details?id=ECLI:NL:RBDHA:2020:865 (2020). ECLI: NL: RBDHA: 2020:865 (NL) and ECLI:NL:RBDHA:2020:1878 (EN) (SyRI): [6.5]
25. Dagar, D., Vishwakarma, D.K.: A literature review and perspectives in deepfakes: generation, detection, and application. Int. J. Multimed Inf. Retr. **11**, 219–289 (2022). https://doi-org.vu-nl.idm.oclc.org/10.1007/s13735-022-00241-w
26. Dalvi, A., Siddavatam, I., Patel, A., Panchal, A., Kazi, F., Bhirud, S.: Predicting attribute effectiveness using biased databases. In: 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), pp. 1–8 (2021). https://doi.org/10.1109/SMARTGENCON51891.2021.9645789
27. Dhami, M.K., Mandel, D.R.: Words or numbers? Communicating probability in intelligence analysis. Amer. Psychol. **76**(3), 549–560 (2021). https://doi.org/10.1037/amp0000637
28. Dias, L.C., Morton, A., Quigley, J.: Elicitation. The Science and Art of Structuring Judgement. International Series in Operations Research & Management Science, vol. 261. Springer, Berlin (2018)
29. Durbach, I.N., Stewart, T.J.: An experimental study of the effect of uncertainty representation on decision making. Eur. J. Oper. Res. **214**, 380–392 (2011). https://doi.org/10.1016/j.ejor.2011.04.021
30. Council of Europe: The convention for the protection of individuals with regard to automatic processing of personal data (cets no. 108). https://www.coe.int/en/web/data-protection/convention108-and-protocol (1981). Accessed 18 Jun 2023
31. Eades III, H., Gadyatskaya, O.: Graphical models for security. In: 7th International Workshop, GraMSec 2020 (2020)
32. Fischhoff, B., Davis, A.L.: Communicating scientific uncertainty. Proc. Natl. Acad. Sci. **111**(Supplement_4), 13664–13671 (2014). https://doi.org/10.1073/pnas.1317504111. https://www.pnas.org/doi/abs/10.1073/pnas.1317504111
33. Frey, S., Rashid, A., Anthonysamy, P., Pinto-Albuquerque, M., Naqvi, S.A.: The good, the bad and the ugly: a study of security decisions in a cyber-physical systems game. IEEE Trans. Softw. Eng. **45**(5), 521–536 (2019). https://doi.org/10.1109/TSE.2017.2782813
34. Friedman, J.A., Zeckhauser, R.: Uncertainty in intelligence. Intell. Natl. Secur. **27**(6), 824–847 (2012). https://doi.org/10.1080/02684527.2012.708275
35. Friedman, J.A., Lerner, J.S., Zeckhauser, R.: Behavioral consequences of probabilistic precision: experimental evidence from national security professionals. Int. Organiz. **71**(4), 803–826 (2017). https://doi.org/10.1017/S0020818317000352
36. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, pp. 1050–1059. JMLR.org (2016)
37. Garae, J., Ko, R.: Visualization and data provenance trends in decision support for cybersecurity. In: Carrascosa, I.P., Kalutarage, H., Huang, Y. (eds.) Data Analytics and Decision Support for Cybersecurity. Springer, Berlin (2017). https://doi.org/10.1007/978-3-319-59439-2_9
38. Gomez, M.: Sound the alarm! updating beliefs and degradative cyber operations. Eur. J. Int. Secur. **4**(2), 190–208 (2019). https://doi.org/10.1017/eis.2019.2
39. Gomez, M.A.: Past behavior and future judgements: seizing and freezing in response to cyber operations. J. Cybersecur. **5** (2019). https://doi.org/10.1093/cybsec/tyz012

40. Gonin, M., Palazzo, G., Hoffrage, U.: Neither bad apple nor bad barrel: how the societal context impacts unethical behavior in organizations. Busin. Ethics Eur. Rev. **21**(1), 31–46 (2012). https://doi.org/10.1111/j.1467-8608.2011.01643.x
41. Groth, P., Moreau, L.: An overview of the prov family of documents. W3C Working Group Note (2013). http://www.w3.org/TR/2013/NOTE-prov-overview-20130430/
42. Henderson, T.C., Simmons, R., Sacharny, D., Mitiche, A., Fan, X.: A probabilistic logic for multi-source heterogeneous information fusion. In: 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Korea (South), pp. 530–535 (2017). https://doi.org/10.1109/MFI.2017.8170375
43. Holzinger, A., Saranti, A., Molnar, C., Biecek, P., Samek, W.: Explainable AI methods – a brief overview. In: Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, K.R., Samek, W. (eds.) xxAI - Beyond Explainable AI. xxAI 2020. Lecture Notes in Computer Science, vol. 13200. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-04083-2_2
44. Hong, J.B., Kim, D.S., Chung, C.J., Huang, D.: A survey on the usability and practical applications of graphical security models. Comput. Sci. Rev. **26**, 1–16 (2017)
45. Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach. Learn. **110**, 457–506 (2021)
46. Irwin, D., Mandel, D.R.: Improving information evaluation for intelligence production. Intell. Natl. Secur. **34**(4), 503–525 (2019). https://doi.org/10.1080/02684527.2019.1569343
47. Irwin, D., Mandel, D.R.: Communicating uncertainty in national security intelligence: expert and nonexpert interpretations of and preferences for verbal and numeric formats. Risk Analysis (2022). https://doi.org/10.1111/risa.14009
48. Janssen, M., der Hoven, J.V.: Big and open linked data (bold) in government: a challenge to transparency and privacy? Govern. Inf. Quart. **32**, 363–368 (2015)
49. Jaspersen, J.G., Montibeller, G.: Probability elicitation under severe time pressure: a rank-based method. Risk Analy. **35**(7), 1317–1335 (2015)
50. Jensen, M.A.: Intelligence failures: what are they really and what do we do about them? Intell. Natl. Secur. **27**(2), 261–282 (2012). https://doi.org/10.1080/02684527.2012.661646
51. Johnson, C.K., Gutzwiller, R.S., Ferguson-Walter, K.J., Fugate, S.J.: A cyber-relevant table of decision making biases and their definitions. Technical Report (2020). https://doi.org/10.13140/RG.2.2.14891.87846
52. Johnson, S.G.B., Merchant, T., Keil, F.C.: Belief digitization: do we treat uncertainty as probabilities or as bits? J. Exper. Psychol. General **149**, 1417–1434 (2020). https://doi.org/10.1037/xge0000720
53. Kahneman, D., Klein, G.: Conditions for intuitive expertise: a failure to disagree. Amer. Psychol. **64**(6), 515–526 (2009). https://doi.org/10.1037/a0016755
54. Kahneman, D., Slovic, P., Tversky, A. (eds.): Judgment under Uncertainty: Heuristics and Biases. Cambridge University Press, Cambridge (1982). https://doi.org/10.1017/CBO9780511809477
55. Karvetski, C.W., Mandel, D.R., Irwin, D.: Improving probability judgment in intelligence analysis: from structured analysis to statistical aggregation. Risk Analy. **40**(5), 1040–1057 (2020). https://doi.org/10.1111/risa.13443
56. Keith, A.J., Ahner, D.K.: A survey of decision making and optimization under uncertainty. Ann. Oper. Res. **300**, 319–353 (2021). https://doi.org/10.1007/s10479-019-03431-8
57. Korff, D., Wagner, B., Powles, J.E., Avila, R., Buermeyer, U.: Boundaries of law: exploring transparency, accountability, and oversight of government surveillance regimes. Cybersecurity (2017)
58. Kowalski, M.: Ethics of Counterterrorism. Boom uitgevers Amsterdam (2017)
59. Labunets, K., Massacci, F., Paci, F.: On the equivalence between graphical and tabular representations for security risk assessment. In: Proceeings of the REFSQ'2016, pp. 191–208 (2017)
60. Laskey, K.J., Laskey, K.B., Costa, P.C.G., Kokar, M.M., Martin, T., Lukasiewicz, T.: Uncertainty reasoning for the world wide web. W3C Incubator Group Report (2008). https://www.w3.org/2005/Incubator/urw3/XGR-urw3-20080331/

61. Li, Y., Chen, J., Feng, L.: Dealing with uncertainty: a survey of theories and practices. IEEE Trans. Knowl. Data Eng. **25**(11), 2463–2482 (2012)
62. Li, V.G., Dunn, M., Pearce, P., McCoy, D., Voelker, G.M., Savage, S.: Reading the tea leaves: A comparative analysis of threat intelligence. In: 28th USENIX Security Symposium (USENIX Security 19), pp. 851–867. USENIX Association, Santa Clara (2019). https://www.usenix.org/conference/usenixsecurity19/presentation/li
63. Logg, J.M., Minson, J.A., Moore, D.A.: Algorithm appreciation: people prefer algorithmic to human judgment. Organiz. Behavior Human Decis. Proc. **151**, 90–103 (2019). https://doi.org/10.1016/j.obhdp.2018.12.005
64. Maathuis, C., Pieters, W., van den Berg, J.: Decision support model for effects estimation and proportionality assessment for targeting in cyber operations. Defence Technol. **17**(2), 352–374 (2021). https://doi.org/10.1016/j.dt.2020.04.007
65. Mandel, D.R.: Assessment and communication of uncertainty in intelligence to support decision-making. NATO STO TECHNICAL REPORT, TR-SAS-114 (2020)
66. Mandel, D.R., Irwin, D.: Facilitating sender-receiver agreement in communicated probabilities: is it best to use words, numbers or both? Judg. Decis. Making **16**(2), 363–393 (2021). https://doi.org/10.1017/S1930297500008603
67. Marlin, B.M., Abdelzaher†, T., Ciocarlie, G., Cobb, A.D., Dennison, M., Jalaian, B., Kaplan, L., Raber, T., Raglin, A., Sharma, P.K., Srivastava, M., Trout, T., Vadera, M.P., Wigness, M.: On uncertainty and robustness in large-scale intelligent data fusion systems. In: IEEE Second International Conference on Cognitive Machine Intelligence (CogMI), pp. 82–91 (2020). https://doi.org/10.1109/CogMI50398.2020.00020
68. Maymí, F.J., Thomson, R.: Human-machine teaming and cyberspace. In: Schmorrow, D., Fidopiastis, C. (eds.) Augmented Cognition: Intelligent Technologies, vol. 10915. Springer, Berlin (2018). https://doi.org/10.1007/978-3-319-91470-1_25
69. Medianovskyi, K., Pietarinen, A.V.: On explainable ai and abductive inference. Philosophies **7**(2), 35 (2022). https://doi.org/10.3390/philosophies7020035
70. Menkveld, C.: Understanding the complexity of intelligence problems. Intell. Natl. Secur. **36**(5), 621–641 (2020). https://doi.org/10.1080/02684527.2021.1881865
71. Montibeller, G., von Winterfeldt, D.: Individual and group biases in value and uncertainty judgments. In: Dias, L.C., Morton, A., Quigley, J. (eds.) Elicitation: The Science and Art of Structuring Judgement, vol. 261, pp. 377–392. Springer, Cham (2018)
72. Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M., Seifert, C.: From anecdotal evidence to quantitative evaluation methods: a systematic review on evaluating explainable AI. ACM Comput. Surv. **55** (2023). https://doi.org/10.1145/3583558
73. Nisioti, A., Loukas, G., Laszka, A., Panaousis, E.: Data-driven decision support for optimizing cyber forensic investigations. IEEE Trans. Inf. Forens. Secur. **16**, 2397–2412 (2021). https://doi.org/10.1109/TIFS.2021.3054966
74. Nunes, I., Jannach, D.: A systematic review and taxonomy of explanations in decision support and recommender systems. User Model. User-Adapted Interact. **27**(3), 393–444 (2017)
75. Okoli, J.O., Weller, G., Watt, J.: Information processing and intuitive decision-making on the fireground: towards a model of expert intuition. Cogn. Tech. Work **18**, 89–103 (2016). https://doi.org/10.1007/s10111-015-0348-9
76. OTAN, N.: Automation in the intelligence cycle (2020). https://www.sto.nato.int/Lists/STONewsArchive/displaynewsitem.aspx?ID=552. Accessed 11 April 2023
77. Padilla, L., Kay, M., Hullman, J.: Uncertainty visualization. In: Piegorsch, W., Levine, R., Zhang, H., Lee, T. (eds.) Computational Statistics in Data Science, pp. 405–421. Wiley, Hoboken (2022)
78. Pagano, T.P., Loureiro, R.B., Lisboa, F.V.N., Cruz, G.O.R., Peixoto, R.M., de Sousa Guimarães, G.A., dos Santos, L.L., Araujo, M.M., Cruz, M., de Oliveira, E.L.S., Winkler, I., Nascimento, E.G.S.: Bias and unfairness in machine learning models: A systematic literature review. arXiv:2202.08176 (2022). https://doi.org/10.48550/arXiv.2202.08176

79. Pawlinski, P., Jaroszewski, P., Kijewski, P., Siewierski, L., Jacewicz, P., Zielony, P., Zuber, R.: Actionable information for security incident response. European Union Agency for Network and Information Security (2014)
80. Perry, W.L., McInnis, B., Price, C.C., Smith, S.C., Hollywoon, J.S.: Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations. Rand Corporation, Santa Monica (2013)
81. Petersen, K.L., Tjalve, V.S.: Intelligence expertise in the age of information sharing: public–private 'collection' and its challenges to democratic control and accountability. Intell. Natl. Secur. **33**(1), 21–35 (2018). https://doi.org/10.1080/02684527.2017.1316956
82. Prabhudesai, S., Yang, L., Asthana, S., Huan, X., Liao, Q.V., Banovic, N.: Understanding uncertainty: How lay decision-makers perceive and interpret uncertainty in human-AI decision making. In: Proceedings of the 28th International Conference on Intelligent User Interfaces, pp. 379–396. Association for Computing Machinery, New York (2023). https://doi.org/10.1145/3581641.3584033
83. Rajivan, P., Cooke, N.J.: Information-pooling bias in collaborative security incident correlation analysis. Human Factors **60**, 626–639 (2018). https://doi.org/10.1177/0018720818769249
84. Ranade, P., Piplai, A., Mittal, S., Joshi, A., Finin, T.: Generating fake cyber threat intelligence using transformer-based models. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–9 (2021). https://doi.org/10.1109/IJCNN52387.2021.9534192
85. Reagans, R., Argote, L., Brooks, D.: Individual experience and experience working together: predicting learning rates from knowing who knows what and knowing how to work together. Manag. Sci. **51**(6), 869–881 (2005). https://doi.org/10.1287/mnsc.1050.0366
86. Regan, H.M., Colyvan, M., Burgman, M.A.: A taxonomy and treatment of uncertainty for ecology and conservation biology. Ecol. Appl. **12**(2), 618–628 (2002). https://doi.org/10.1890/1051-0761(2002)012[0618:ATATOU]2.0.CO;2
87. Rona-Tas, A., Cornuéjols, A., Blanchemanche, S., Duroy, A., Martin, C.: Enlisting supervised machine learning in mapping scientific uncertainty expressed in food risk analysis. Sociol. Methods Res. **48**(3), 608–641 (2019)
88. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson, Saddle River (2020)
89. Slayton, R.: What is the cyber offense-defense balance? Conceptions, causes, and assessment. Int. Secur. **41**(3), 72–109 (2017). https://doi.org/10.1162/ISEC_a_00267
90. Snidaro, L., Visentini, I., Bryan, K.: Fusing uncertain knowledge and evidence for maritime situational awareness via markov logic networks. Inf. Fusion **21**, 159–172 (2015). https://doi.org/10.1016/j.inffus.2013.03.004. https://www.sciencedirect.com/science/article/pii/S1566253513000523
91. Stevens, R., Votipka, D., Redmiles, E.M., Ahern, C., Sweeney, P., Mazurek, M.L.: The battle for New York: A case study of applied digital threat modeling at the enterprise level. In: 27th USENIX Security Symposium, pp. 621–63 (2018)
92. Tounsi, W., Rais, H.: A survey on technical threat intelligence in the age of sophisticated cyber attacks. Comput. Secur. **72**, 212–233 (2018). https://doi.org/10.1016/j.cose.2017.09.001
93. Tuma, K., Scandariato, R.: Two architectural threat analysis techniques compared. In: Software Architecture: 12th European Conference on Software Architecture, ECSA 2018, Madrid, Spain, September 24–28, 2018, Proceedings 12, pp. 347–363. Springer, Berlin (2018)
94. Tuma, K., Van Der Lee, R.: The role of diversity in cybersecurity risk analysis: An experimental plan. In: 3rd Workshop on Gender Equality, Diversity, and Inclusion in Software Engineering, GEICSE 2022, pp. 12–18. Institute of Electrical and Electronics Engineers (2022)
95. Tuma, K., Calikli, G., Scandariato, R.: Threat analysis of software systems: a systematic literature review. J. Syst. Softw. **144**, 275–294 (2018)

96. Tuma, K., Sion, L., Scandariato, R., Yskout, K.: Automating the early detection of security design flaws. In: Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 332–342 (2020)
97. Tuma, K., Sandberg, C., Thorsson, U., Widman, M., Herpel, T., Scandariato, R.: Finding security threats that matter: Two industrial case studies. J. Syst. Softw. **179**, 111003 (2021)
98. van der Kleij, R., Schraagen, J.M., Cadet, B., Young, H.: Developing decision support for cybersecurity threat and incident managers. Comput. Secur. **113**, 102535 (2022). https://doi.org/10.1016/j.cose.2021.102535
99. van der Voort, H., Klievink, A., Arnaboldi, M., Meijer, A.: Rationality and politics of algorithms. will the promise of big data survive the dynamics of public decision making? Govern. Inf. Quart. **36**(1), 27–38 (2019). https://doi.org/10.1016/j.giq.2018.10.011
100. Villiers, J.P.D., Laskey, J.P., Jousselme, A., Blasch, E., de Waal, A., Pavlin, G., Costa, P.: Uncertainty representation, quantification and evaluation for data and information fusion. In 2015 18th International Conference on Information Fusion. IEEE, pp. 50–57 (2015)
101. Villiers, J.P.D., Pavlin, G., Jousselme, A., Maskell, S., de Waal, A., Laskey, K., Costa, P., Blasch, E.: Uncertainty representation and evaluation for modeling and decision-making in information fusion. J. Adv. Inf. Fusion **13**, 198–215 (2018)
102. Vogel, K.M., Reid, G., Kampe, C., Jones, P.: The impact of ai on intelligence analysis: tackling issues of collaboration, algorithmic transparency, accountability, and management. Intell. Natl. Secur. **36**(6), 827–848 (2021). https://doi.org/10.1080/02684527.2021.1946952
103. Waardenburg, L., Sergeeva, A., Huysman, M.: Hotspots and blind spots. In: Schultze, U., Aanestad, M., Mähring, M., Østerlund, C., Riemer, K. (eds.) Living with Monsters? Social Implications of Algorithmic Phenomena, Hybrid Agency, and the Performativity of Technology, pp. 96–109. Springer International Publishing, Cham (2018)
104. Wagner, T.D., Mahbub, K., Palomar, E., Abdallah, A.E.: Cyber threat intelligence sharing: survey and research directions. Comput. Secur. **87** (2019). https://doi.org/10.1016/j.cose.2019.101589
105. Wei, L., Du, H., Mahesar, Q.A., Al Ammari, K., Magee, D.R., Clarke, B., Dimitrova, V., Gunn, D., Entwisle, D., Reeves, H., Cohn, A.G.: A decision support system for urban infrastructure inter-asset management employing domain ontologies and qualitative uncertainty-based reasoning. Expert Syst. Appl. **158**, 113461 (2020). https://doi.org/10.1016/j.eswa.2020.113461
106. Whitesmith, M.: The efficacy of ach in mitigating serial position effects and confirmation bias in an intelligence analysis scenario. Intell. Natl. Secur. **34**(2), 225–242 (2019). https://doi.org/10.1080/02684527.2018.1534640
107. Whyte, C.: Learning to trust skynet: Interfacing with artificial intelligence in cyberspace. Contempor. Secur. Policy **44**(2), 308–344 (2023). https://doi.org/10.1080/13523260.2023.2180882
108. Willingham, D.T., Riener, C.: Cognition: The Thinking Animal, 4th edn. Cambridge University Press, Cambridge (2019). https://doi.org/10.1017/9781316271988
109. Wirtz, J.J.: The sources and methods of intelligence studies. In: Johnson, L.K. (ed.) The Oxford Handbook of National Security Intelligence. Oxford University Press, Oxford (2010). https://doi.org/10.1093/oxfordhb/9780195375886.003.0004
110. Wu, J., Li, H.: Uncertainty analysis in ecological studies: An overview. In: Wu, J., Jones, K.B., Li, H., Loucks, O.L. (eds.) Scaling and Uncertainty Analysis in Ecology, pp. 45–66. Springer Netherlands, Dordrecht (2006). https://doi.org/10.1007/1-4020-4663-4_3
111. Xiong, W., Lagerström, R.: Threat modeling – a systematic literature review. Comput. Secur. **84**, 53–69 (2019). https://doi.org/10.1016/j.cose.2019.03.010

112. Zhao, K., Li, L., Chen, Z., Sun, R., Yuan, G., Li, J.: A survey: optimization and applications of evidence fusion algorithm based on dempster–shafer theory. Appl. Soft Comput. **124**, 109075 (2022). https://www.sciencedirect.com/science/article/pii/S1568494622003696. https://doi.org/10.1016/j.asoc.2022.109075
113. Zibak, A., Sauerwein, C., Simpson, A.C.: Threat intelligence quality dimensions for research and practice. Digital Threats Res. Practice **3**(4), 44 (2022). https://doi.org/10.1145/3484202

Securing the Future: The Role of Knowledge Discovery Frameworks

Martins Jansevskis and Kaspars Osis

1 Preword

In the DIKW (Data, Information, Knowledge, Wisdom) hierarchy, data represents raw facts and figures devoid of context or meaning, information is structured data with context and relevance, and knowledge encompasses synthesized and contextualized information that is actionable and useful (see Fig. 1) [23].

Knowledge discovery incorporates machine learning as a component within a broader system. These technologies are harnessed to process and analyze large datasets, allowing organizations to extract actionable insights and enhance their decision-making processes.

2 Knowledge Discovery Frameworks

A knowledge society distinguishes between information and knowledge, emphasizing the identification, creation, processing, transformation, distribution, and utilization of information to generate and apply knowledge [1]. Knowledge discovery refers to the process of extracting insights and understanding from information through the use of specific techniques. The extraction of knowledge from data, as provided by intelligent systems, increasingly relies on the incorporation of machine learning algorithms and artificial intelligence techniques, prompting a growing need for knowledge discovery frameworks and methods that address escalating security concerns [2]. To facilitate the development of knowledge discovery systems, knowl-

M. Jansevskis (✉) · K. Osis
Vidzeme University of Applied Sciences, Valmiera, Latvia
e-mail: martins.jansevskis@va.lv; kaspars.osis@va.lv

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_5

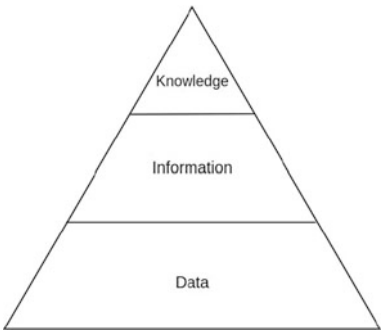


Fig. 1 DIKW pyramid, excluding wisdom. (Adapted from [23])

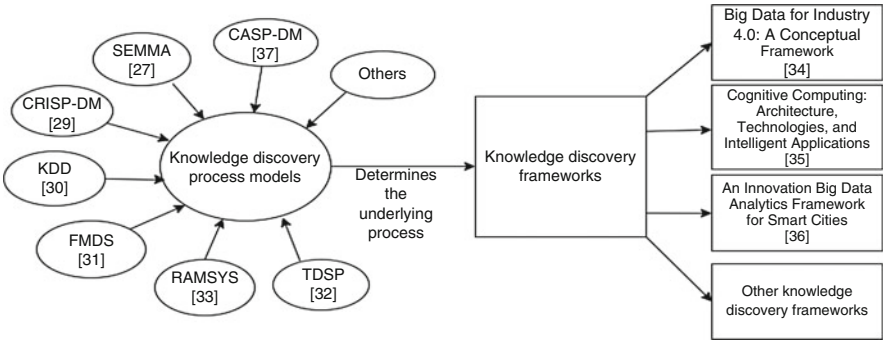


Fig. 2 Knowledge discovery process models in the context with knowledge discovery frameworks. (Adapted from [3])

edge discovery frameworks combine various techniques and tools from different fields, such as machine learning, data mining, data visualization, statistics, and knowledge discovery process models. These frameworks play a role in assisting researchers and practitioners in uncovering patterns and relationships within data that enhance understanding and decision-making. As the complexity of knowledge discovery systems increases, the use of these frameworks, process models, and design patterns becomes increasingly important. By harnessing these tools, individuals and organizations can streamline the development process, enhance efficiency, and reduce the risk of errors [3, 24].

Process models are a popular approach to developing knowledge discovery systems, with some of the earliest models, such as Knowledge Discovery in Databases (KDD) and CRoss Industry Standard Process for Data Mining (CRISP-DM), emerging in the late 1990s (see Fig. 2) [3].

As knowledge discovery has advanced, these models have been refined and integrated into broader knowledge discovery frameworks. While newer models, such as SEMMA (Sample, Explore, Modify, Model, and Assess) and Team Data Science Process (TDSP), have been introduced, CRISP-DM remains widely used and is considered a standard for data acquisition and retrieval projects [4, 5]. However, it is essential to note that no single process model or framework is

universally applicable and appropriate for all knowledge discovery projects (see Fig. 2). Furthermore, developing a knowledge discovery system should be flexible and adaptive, incorporating best practices and tools from various process models and frameworks [3].

Authors have researched existing knowledge discovery frameworks [3], and the frameworks have been chosen so that each is for a different domain (see Table 1). The frameworks are innovative, and their presenting publications have been used in related research. The criteria are established to ensure framework selection: (1) the knowledge discovery framework is not older than 6 years, and (2) the publication representing the framework, according to Scopus Field-Weighted Citation Impact, is rated with a value of at least 9, which means that this publication has added value in the domain. Based on the criteria, three frameworks were analyzed in-depth: "An Innovative Big Data Analytics Framework for Smart Cities" [36], "Cognitive Computing: Architecture, Technologies, and Intelligent Applications," [35] and "Big Data for Industry 4.0: A Conceptual Framework" [34].

Table 1 shows that all of the compared frameworks are based on modular design and include results processing and distribution modules: the characteristics can be considered a requirement for a framework. The high-performance, scalable infrastructure and abstraction from the complexity of data processing platforms, distributed infrastructure, and preprocessing modules are also present in two of three frameworks. However, none of the compared frameworks include security and legislations as part of a framework.

Knowledge discovery frameworks are developed to serve specific objectives (see Table 1), whether in research, academia, or practical applications, and may encompass infrastructure definitions, modules, components, and technology stacks.

Table 1 Knowledge discovery frameworks feature comparison, adapted from [3]

Feature	Industry 4.0 conceptual framework	Cognitive computing framework	Framework for smart cities
Has a description of the implementation	X	X	✓
Modular design	✓	✓	✓
Developers abstracted from the complexity of data processing platforms	✓	X	✓
Distributed infrastructure	✓	X	✓
Data preprocessing module	✓	X	✓
Results processing and distribution module	✓	✓	✓
Connectable programming nodes	✓	X	X
High-performance scalable infrastructure	✓	X	✓
Support for big data platforms	X	X	✓
Potential technologies and their application are presented	X	X	✓
Online analytics module	X	X	✓
Security and legislations as part of framework	X	X	X

Knowledge discovery process models are essential to such frameworks, as they outline the underlying process of discovering knowledge (see Fig. 2) [3]. By utilizing a process model, organizations can gain a clear understanding of the steps involved in knowledge discovery and the roles and responsibilities of team members at each stage. This helps ensure that the knowledge discovery process is efficient, effective, and aligned with the project's needs.

Data can often have multiple applications in fields not directly related to the domain where it was initially collected. Models of the knowledge discovery process provide a structured and systematic approach for uncovering patterns and insights in complex datasets. As a result, a comprehensive knowledge discovery framework can include different process model approaches (see Fig. 2). This underscores the importance of having a flexible and adaptable framework that can accommodate evolving data requirements and objectives [3].

Discovering factual knowledge involves a series of tasks, including problem definition, framework and model development, and evaluation. Knowledge discovery systems are designed to meet various objectives and can be easily integrated or linked with existing processes and systems. However, the adoption of knowledge discovery frameworks in various settings is influenced by various concerns, such as compatibility with existing systems and processes and data privacy and security considerations [6].

Security is a critical concern for knowledge discovery systems as they are designed to extract valuable insights and information from large datasets, making them potential targets for malicious actors seeking to compromise sensitive information. With adequate security measures, knowledge discovery systems can be protected from various cyber threats, including data breaches, theft, and misuse. Therefore, it is essential to prioritize security when developing and deploying knowledge discovery systems to ensure they operate effectively and protect data confidentiality, integrity, and availability [25]. By implementing robust security protocols and leveraging the latest technologies and best practices, organizations can minimize the risks associated with knowledge discovery and maximize the value of their insights [7].

3 Knowledge Discovery Systems Constraints

Knowledge discovery systems are impacted by a multitude of factors, shaping the landscape in which these systems operate. Among the influences are the various constraints imposed by legal frameworks, privacy regulations, and the ever-evolving field of cybersecurity. These constraints not only govern the manner in which data is collected, processed, and disseminated but also establish the ethical and legal boundaries that knowledge discovery systems must navigate. In this environment, it becomes increasingly imperative to achieve a balance between the insights and the imperative to respect the rights and security of individuals and organizations.

The advent of big data technologies has brought a transformation in knowledge discovery systems, providing economic benefits. The knowledge derived from these systems plays a role in shaping strategies for social and economic development. The emerging big data service architecture represents an economy rooted in services, which places a premium on data as a valuable resource by gathering and extracting information from diverse sources [8]. The synergy between big data and knowledge discovery systems empowers personalized data processing, analysis, and visualization services, facilitating informed decision-making. The service architecture typically comprises either three (application, processing, collection, and storage [8]) or five primary layers (collection, storage, processing, analytics, and application [9]). Nevertheless, enabling technologies underpin these layers, facilitating information collection, management, analysis, and visualization to catalyze knowledge discovery. Consequently, organizations can harness these technologies to gain insights into their information, elevate decision-making processes, and achieve their objectives.

In the realm of knowledge discovery systems, data storage splits into batch and dynamically streamed data. Batch data denotes data stored in a static format, while streamed data entails a continuous flow of real-time data records [10]. Handling streamed data in knowledge discovery systems necessitates solutions characterized by prompt operations, fault tolerance, stability, and reliability. Conversely, batch processing is conventionally managed using tools like Hadoop and MapReduce. For processing streaming data, popular choices encompass Storm, Spark, and Samza [8, 9].

In recent years, the increase of data has created new demands in data storage, prompting widespread adoption of distributed file systems, including NoSQL, NewSQL, and other data management systems. Notably, the Hadoop Distributed File System (HDFS) has emerged as a predominant choice for large-scale data storage, offering redundancy and scalability within parallel distributed architecture systems [11].

NoSQL databases have been crafted to cater to operational requirements, particularly in real-time applications. NoSQL provides rapid and efficient processing of vast data volumes by storing data in an unstructured format across multiple processing nodes and servers. Consequently, the distributed database infrastructure of NoSQL is deemed highly conducive to knowledge discovery systems harboring extensive data repositories [12]. Moreover, NoSQL databases provide advantages such as scalability, flexibility, and fault tolerance, making them a good option for applications and reliable data processing. The utilization of NoSQL databases demands planning and consideration, as their unstructured nature can pose challenges related to data consistency and security [7].

Cloud computing technology underscores distributed processing capabilities, distributed databases, and virtualization technologies as its key benefits. According to scholarly literature, the integration of cloud computing platforms represents a pivotal element in the development of knowledge discovery systems [8, 9]. The three primary cloud computing architectures encompass SaaS (Software as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service) [26]. The

adoption of any of these architectural models allows organizations to drop the necessity for maintaining in-house server infrastructure or crafting solutions, as cloud services are readily accessible to fulfill specific tasks [7].

Prominent cloud service providers such as Amazon Web Services, Microsoft Azure, and Google Cloud Platform are known for their comprehensive offerings. They furnish infrastructure capable of accommodating a broad spectrum of information development requirements, capable of supporting the entire life cycle of knowledge discovery system creation and data processing. Ample computing power is indispensable not solely for technical processes but also for data processing. While it is feasible to develop, maintain, and deliver tailored solutions with limited developer or customer resources, such an approach can amplify the intricacy of the technical processes [7].

In addition to the technical requirements encompassing information storage, processing, visualization, and application of cloud computing resources, knowledge discovery systems must conform to legal standards governing data collection and storage. Moreover, these systems must encompass preventive and corrective measures to uphold cybersecurity.

Knowledge discovery systems are created to extract valuable insights from information and accordingly data through the processes of collection, processing, and analysis. However, it is necessary to take into account regulations concerning the protection of personal data, such as GDPR or US Privacy Act, which mandate the collection and use of only the minimum essential data for specific purposes [13]. Organizations are obliged to assess whether their utilization of personal data is reasonable and aligns with data subject expectations. The accumulation of substantial data volumes can potentially conflict with these principles [14]. Therefore, knowledge discovery systems must adhere to principles governing personal data protection concerning data storage, analysis, and utilization. These principles ensure that personal data is safeguarded, processed fairly and lawfully, and retained no longer than necessary, as enshrined in the Charter of Fundamental Rights of the European Union and GDPR.

Compliance with GDPR can be achieved through data anonymization or pseudonymization within knowledge discovery systems. Anonymization involves rendering data unidentifiable, thereby exempting it from the scope of GDPR [13]. Article 22(1) of GDPR grants individuals the right to abstain from automated decision-making, including profiling, typically underpinned by machine learning algorithms. However, automated decision-making may be permissible when required for executing legal contracts or when data subjects provide explicit consent. Profiling through knowledge discovery systems often eludes individuals' awareness, potentially violating data protection regulations. To uphold individuals' rights, organizations must ensure transparency and provide options for opting out of automated profiling and decision making [14].

Cybersecurity constitutes a concern for knowledge discovery systems and networks, given their escalating significance in daily lives. Cybersecurity encompasses safeguarding hardware, software, electronic data, and services against unauthorized access, theft, damage, or disruption [15]. This is particularly pertinent as more

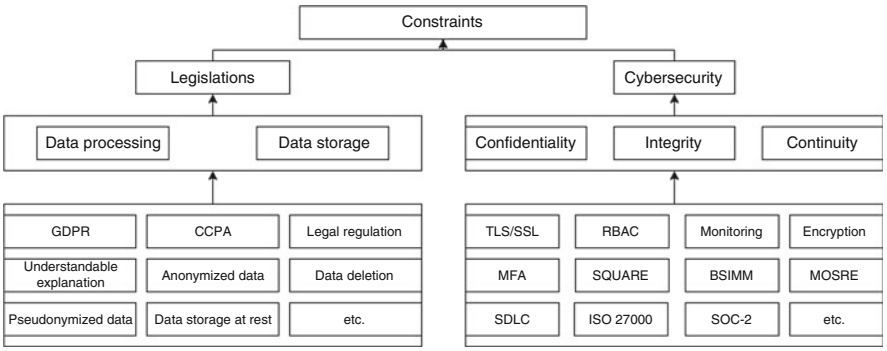


Fig. 3 Knowledge discovery frameworks constraints. (Adapted from [7])

individuals rely on wireless network standards like Bluetooth and Wi-Fi and employ smart devices such as smartphones, televisions, and other Internet of Things (IoT) devices. Owing to the multifaceted nature of cybersecurity, spanning political and technological dimensions, it has burgeoned into one of the foremost challenges [15, 16]. Consequently, it is indispensable for knowledge discovery systems to incorporate cybersecurity requirements into their developmental life cycle and accordingly into knowledge discovery framework, encompassing measures to shield against cyber threats, ensure data privacy, and preserve system integrity.

Access to the knowledge discovery system necessitates authentication and authorization. For highly sensitive data, multifactor authorization (MFA) emerges as a recommended solution (see Fig. 3). Access rights must be defined in accordance with the principle of least privilege. For instance, if a component of the knowledge discovery system, such as sensors or surveillance cameras, requires data storage capabilities, its access rights should exclude privileges related to reading, deleting, or modifying data. This approach is endorsed by numerous researchers [15, 16].

To ensure information security in a knowledge discovery system, adopting encrypted storage both at rest and during transit between system components is necessary. This approach adds an extra layer of protection, safeguarding information in case an unauthorized actor gains access to the dataset. The utilization of the Transport Layer Security (TLS) protocol is highly recommended for encrypting data during transit. Equally critical is the prevention of system components from accessing the repository without TLS, a practice advocated by experts such as Xin et al. [16] and Schatz et al. [15], who emphasize data encryption as a foundational security measure in knowledge discovery systems.

In tandem with data encryption, a proactive process is indispensable to ensure the security of knowledge discovery systems. This proactive approach should encompass a comprehensive monitoring strategy for the knowledge discovery system's infrastructure. The monitoring process should continuously oversee the data storage and computing infrastructure to promptly detect and mitigate potential security threats. Cloud resource monitoring tools, exemplified by Amazon Cloud-

Watch, Azure Monitor, and Google Cloud Operations, offer a valuable means to aggregate monitoring and operational data and create informative metrics. System administrators can harness these tools to gain insights into system health and performance, allowing them to take corrective actions whenever necessary [15, 16].

Intrusion detection systems (IDSs) represent a cornerstone of any robust cybersecurity strategy. These systems are engineered to identify and respond to potential threats within a computer system or network [17, 18]. IDSs operate by scrutinizing network traffic and system activity, adeptly discerning suspicious behavior, and alerting system administrators about the potential occurrence of an attack. This early warning mechanism serves to preemptively avert or minimize the impact of a security breach. Beyond threat detection, IDS can also illuminate vulnerabilities in a system's security posture, offering valuable insights for fortifying defenses. The implementation of IDS is necessary for organizations entrusted with sensitive data or serving a substantial user base. By integrating IDS, it becomes significantly more manageable to detect and respond to potential threats. Overall, investing in an intrusion detection system is indispensable for preserving the security and integrity of knowledge discovery systems and networks [17, 18].

Intrusion detection systems (IDSs) hold a role in upholding the security of knowledge discovery systems. Traditional IDSs hinge on signature-based methodologies for detecting known attacks, which may be constrained when confronted with novel and evolving threats. However, the advent of deep learning-based IDSs has exhibited promise in the detection of previously unseen attacks, boasting high accuracy levels [17]. These systems harness neural networks to acquire insights into patterns of normal network behavior, enabling them to identify deviations and anomalies from this norm. By leveraging deep learning-based IDSs, knowledge discovery systems can adeptly detect and respond to potential security threats in real time, thereby mitigating the risk of data compromise or loss [17]. This assumes added significance in light of the escalating complexity of cyberattacks and the high volume of sensitive data entrusted to knowledge discovery systems. By incorporating deep learning-based IDSs into their security arsenal, organizations can augment the protection of their knowledge discovery systems and prevent security breaches [17, 18].

The development of knowledge discovery systems is impacted by a multitude of constraints and must include specific data protection measures into the system and compact importance to security requirements during the design phase. The landscape of knowledge discovery systems is associated with legal statutes and cybersecurity imperatives, as illustrated in Fig. 3. Compliance with regional laws necessitates the integration of data protection measures and constraints regarding the collection and utilization of personal data. Cybersecurity assumes equal prominence given the increasing reliance of knowledge discovery systems on computer infrastructure, the Internet, and wireless networks. Hence, developers must prioritize security requirements during the knowledge discovery system planning phase and draw upon methodologies such as ISO 27000 and SOC-2 certification. Upholding the safety and integrity of knowledge discovery systems includes the incorporation of data security practices such as the principle of least privilege, data encryption, continuous monitoring, and a judicious reduction in the attack surface. Developers

can create effective and compliant knowledge discovery systems by applying architectures with security at their core while comprehending the flow of data within the infrastructure [7].

4 Knowledge Discovery Framework Proposal

Knowledge discovery systems present high complexity, stemming from both their technological foundations and the cybersecurity prerequisites essential for ensuring their safe and secure operation [7]. As such, the development of knowledge discovery systems demands attention to the technological and cybersecurity factors. Frameworks, serving as conceptual abstractions, outline the composition of components that can be tailored in alignment with specific requirements, thereby providing distinct operational characteristics. However, the existing pool of frameworks available for knowledge discovery system development frequently does not include cybersecurity considerations that necessitate greater adaptability across multiple sectors [3]. Consequently, the selection of a framework mandates the assessment of technological and cybersecurity prerequisites, ensuring alignment with the requirements of the knowledge discovery system under construction.

The lack of knowledge discovery frameworks that can be used in different areas makes it take longer to develop systems and highlights how important security is. To address this challenge, the authors introduce a conceptual framework titled the User Interaction and Response-based Knowledge Discovery Framework (UIS-KDF) (see Fig. 4) [7].

Proposed framework delineates five logical layers, namely, public applications, management applications, machine learning, constraints (detailed upon in the preceding section), and technology (see Fig. 4). The UIS-KDF framework is a meta level designed for the needs of knowledge discovery systems. By embedding security considerations throughout the phases of design, development, and deployment, the UIS-KDF framework addresses the vulnerabilities associated with data breaches, cyber intrusions, and security challenges.

At the core of the UIS-KDF framework is a technology layer, designed to accommodate multiple data formats including structured, semi-structured, and unstructured data. To safeguard the integrity and confidentiality of data, the UIS-KDF framework encompasses a layer of constraints (defense line), including an array of security measures categorized into cybersecurity and legislative subdomains. These measures are included to secure data at rest and in transit, prevent unauthorized access to sensitive information, and align with legal statutes.

In summation, the UIS-KDF framework represents a comprehensive and adaptable approach to knowledge discovery system development, where security considerations are included from the beginning of the development process. By providing organizations with an adaptable and scalable solution, this framework empowers them to utilize the potential within their data and also mitigates the security breaches and cyber threats.

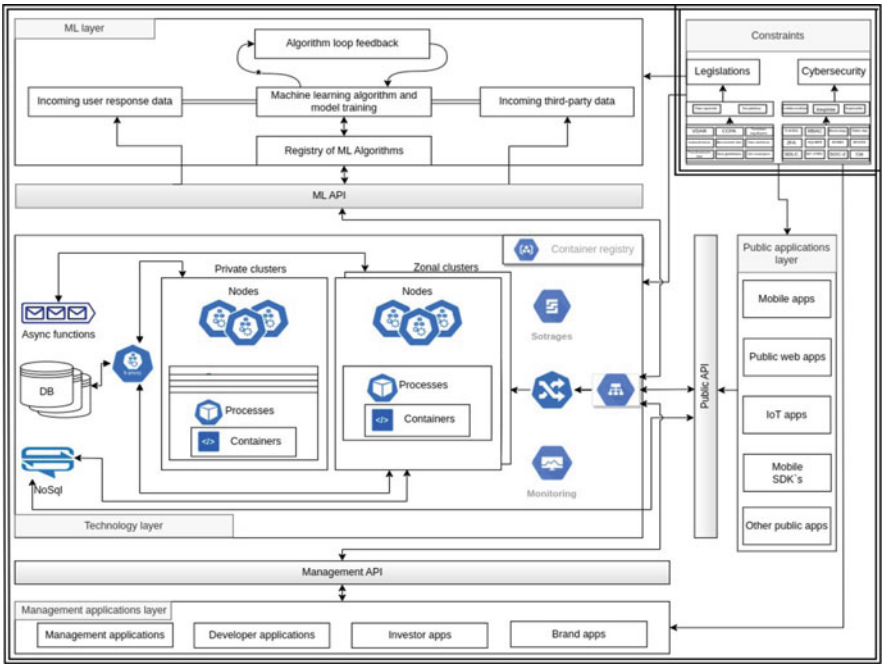


Fig. 4 User Interaction and Response-based Knowledge Discovery Framework (UIS-KDF). (Adapted from [7])

4.1 Private and Public Application Layer

The UIS-KDF framework comprises two distinct application layers: the public application layer and the management application layer (see Fig. 4). These layers serve different purposes and possess different characteristics.

The public application layer is faced toward end users and focuses on delivering essential business functionalities. It interfaces with the technology layer's APIs, providing adaptability through customization, addition, or replacement of public applications. These applications can be web-based, mobile, hybrid, or desktop, supporting a variety of platforms and systems. Whether open source or closed source, they require compatibility with APIs, irrespective of the API technology used (REST, GraphQL, RPC, SOAP).

In contrast, the management application layer consists of applications dedicated to managing internal processes and procedures within the organization. These applications provide specific functionality for a limited number of users and well-defined usage scenarios. Unlike public applications, management applications tend to have fewer users and predictable usage patterns.

The distinction between public and management applications lies in their end users and potential usage loads. Public applications may experience an unlimited

number of users and unpredictable usage patterns, necessitating API development that prioritizes scalability, load management, and robust authentication to handle varying loads effectively. On the other hand, API development for management applications can concentrate on delivering functionality for a smaller user base and clearly defined usage scenarios [7].

Encapsulation is a technique that involves wrapping the implementation details of a component or application and exposing only the necessary interfaces for interaction [22]. This approach offers multiple advantages, including improved usability, reduced complexity, and enhanced security. By encapsulating applications, security measures can be more efficiently applied to shield them from potential threats. It allows for the implementation of access control mechanisms, such as authentication and authorization, to restrict access to authorized users. Furthermore, encapsulation establishes a distinct boundary between the application and other system components, facilitating better isolation and minimizing the impact of potential security breaches. In summary, encapsulation integrated into knowledge discovery framework streamlines application protection by establishing clear boundaries, access limitations, and effective security measures.

The segregation of management and public applications plays a role in achieving the specific objectives of organizations and bolstering their cybersecurity stand. The practice of encapsulating application functionalities simplifies the task of safeguarding them against cyber threats. By isolating and segregating internal management applications from public-facing ones, organizations can control access to sensitive data and reduce the attack surface. This separation also simplifies the implementation of suitable access controls, such as role-based access control (RBAC) and the principle of least privilege. Such measures enhance the overall security of the system and ensure compliance with regulatory requirements, particularly those pertaining to data privacy and protection. Consequently, a well-structured and encapsulated application architecture can enhance an organization's security posture while enabling the efficient and reliable delivery of services to end users [7].

4.2 Technology Layer

The technology layer (see Fig. 4) within the UIS-KDF framework plays a role in delivering functionality and underpinning business processes, thereby facilitating data-driven decision-making. This layer incorporates responsibility for managing an array of critical components, including technology stacks, operating systems, containers, communication protocols, databases, gateways, monitoring and management solutions, and core processes that drive the knowledge discovery system's operations. A core function for this layer is to maintain a clear delineation between technology and the business and application functionality. Layer also incorporates cybersecurity features, such as multilevel user authentication, audit logs, and role-based access control. The technology layer should be architectured to

facilitate scaling of computational resources in response to the system's changing demands [7].

To create a technology layer, the recommended approach is to harness containerization technologies like Docker, LXD, and Containerd, coupled with container orchestration platforms such as Kubernetes, Rancher, Google Cloud Run, and AWS Fargate. Additionally, the technology layer harnesses the capabilities of cloud computing technologies, including distributed computing, distributed databases, virtualization technologies, and automated vertical and horizontal scalability. In this regard, cloud computing services like Amazon Web Services, Microsoft Azure, Google Cloud Platform, or analogous services are embraced. Furthermore, the technology layer must be equipped for dynamic scaling of computational resources, multilevel user authentication, comprehensive audit trail generation, and the implementation of role-based access control [7].

Cloud computing platforms offer a suite of functionalities encompassing data continuity, integrity, and availability, rendering them suitable for storing data for knowledge discovery systems. The choice of cloud storage services has to align with the system's technical requirements and can be hosted on the infrastructures of various service providers. Within the UIS-KDF framework, the role of the cloud computing platform is to furnish an application interface that can be leveraged by applications across the private, public, and machine learning layers. Database solutions and clusters, such as traditional relational database systems (SQL) or contemporary storage solutions (NoSQL) like Apache Flume, HDFS, SQL, and MongoDB, among others, can be implemented to store high volumes of data.

The technology layer encompasses an array of functions, including load balancing management, asynchronous operation management, system image registries, public and management application APIs, monitoring solutions, and a spectrum of supporting technologies. By effecting a clear demarcation between technology and the domains of organizational, application, and machine learning functionality, the technology layer empowers organizations to concentrate on their core objectives while sustaining the requisite technological underpinnings [7]. Recognizing that the management of technology can diverge from the pursuit of organizational goals, allocating technology-related responsibilities to a dedicated layer not only enhances system performance but also mitigates the likelihood of errors.

4.3 Machine Learning Layer

The machine learning (ML) layer stands as an essential component within the UIS-KDF framework, as illustrated in Fig. 4. Its primary function is to facilitate knowledge discovery and oversee the management of machine learning models. By harnessing this layer, organizations can extract insights and do predictions from data, thus empowering data-driven decision-making and delivering enhanced services to end users. Furthermore, it has to be designed to facilitate the deployment and administration of machine learning models. Machine learning platforms like

TensorFlow, PyTorch, Scikit-Learn, and others can significantly aid in achieving these aims. Leveraging machine learning models within knowledge discovery systems brings advantages for organizations, including process automation, resource optimization, and heightened service quality for end users [7].

The machine learning layer, depicted in Fig. 4, consists of several components, comprising an algorithm registry, feedback loops, third-party data integration, and user response data. The algorithm registry assumes the role of accommodating multiple ML algorithms tailored for diverse purposes, thereby augmenting prediction accuracy. Additionally, it simplifies the substitution of algorithms when needed and their seamless integration into a unified system. Feedback loops, on the other hand, are for perpetually advancing and refining machine learning models. They allow the system to acquire insights from past mistakes and subsequently enhance its predictive capabilities. In recent years, the landscape of machine learning has witnessed expansion and refinement, culminating in the development of more sophisticated algorithms that provide better insights and predictive precision, therefore popularizing knowledge discovery systems [19, 27].

The ML layer can also derive benefits from third-party data sources, which can be incorporated to enrich the dataset, elevating the quality of predictions. Data plays an important role in the functioning of the knowledge discovery system, as it enables the identification of patterns and trends in behavior, ultimately contributing to the service quality.

By demarcating the machine learning layer from the public and management application layers, the ML functionality and training process can be isolated, improving management and fortification of the encapsulated applications. This segregation serves to enhance the security and privacy aspects of the system by circumscribing access to sensitive data and functions.

In summation, the machine learning layer constitutes a crucial element within the UIS-KDF framework, affording the functionality for knowledge discovery and the administration of machine learning models. Furthermore, the integration of ML algorithms coupled with continuous learning through feedback loops empowers organizations to automate operations, optimize resource utilization, and elevate the quality of services dispensed to end consumers.

In addition to its role in enhancing service quality and automation, machine learning algorithms can be deployed to detect potential threats, encompassing the identification of anomalous user behavior, detection of fraudulent activities, and flagging of suspicious network traffic. Leveraging machine learning algorithms, organizations can improve their security posture and curtail the risk of cyberattacks [7].

4.4 UIS-KDF

The process of knowledge discovery entails the extraction of valuable knowledge from data through the application of diverse techniques. Its primary objective

revolves around acquiring actionable insights from data that can be harnessed to enhance organizational processes and inform decision-making. This unfolds through the utilization of data mining algorithms, which scrutinize and decipher data based on predefined conditions and specifications [20]. The realms of both academia and industry have created several process models, delineating the various stages constituting the knowledge discovery process. These models typically encompass a range of activities that adhere to the principles of project management [21].

Facilitating the development of knowledge discovery systems hinges on the usage of knowledge discovery frameworks, which provide development directives. These frameworks set out to streamline the intricate and time-intensive process of crafting knowledge discovery systems, particularly in the context of extracting knowledge from data. By applying these frameworks, organizations can build a structure for creating and using knowledge discovery systems that fit their business needs. Knowledge discovery frameworks provide guidance for the process of creating knowledge discovery systems, starting with defining the problem and ending with deploying and monitoring the ML models. Furthermore, these frameworks include best practices and methodologies including data preprocessing, feature engineering, model training, and model selection. Adhering to these guidelines allows organizations to develop knowledge discovery systems that acquire knowledge from data, refine their services, and attain their business objectives [3].

The UIS-KDF framework is designed to be adaptable across multiple sectors, modular or distributed, including the security and legal aspects for organizations looking for a valuable guide to develop knowledge discovery capabilities. It's divided into different layers, including technology, management, public applications, constraints, and machine learning, although they are loosely connected. These layers have specific requirements for development. However, the collaboration between these components is crucial for data-driven decision-making in organizations. The framework provides a clear structure that helps organizations understand how different parts work together to achieve their goals. This leads to better decision-making and efficient use of resources [7].

The UIS-KDF framework furnishes blueprints for architectural schematics and organizational paradigms while elevating the cybersecurity quotient of knowledge discovery systems that harness machine learning and artificial intelligence. With the increase in data and more organizations using machine learning, cybersecurity has become an important factor. The UIS-KDF framework provides a way to separate the machine learning part, which deals with knowledge discovery and training, from the rest of the system. This demarcation serves to curtail the system's prospective attack surface, rendering it more arduous for malicious actors to infiltrate and access sensitive information.

The UIS-KDF framework includes the foundation for protecting data privacy, making sure data is not tampered with and is always available. It also highlights how important things like encryption, access control, and auditing are to defend against malicious actors. By bringing all these factors together, the UIS-KDF framework improves the security of the knowledge discovery system. It does this by separating

machine learning, setting rules to keep data safe, including continuous monitoring, and making data security a top priority in the development process [7].

The framework has demonstrated its helpfulness through successful applications within the advertising and mobile technology sectors. It has proven to be a versatile, adaptable, and valuable knowledge discovery system development guide. Additionally, the UIS-KDF framework is a fruit of a larger project that conducts comprehensive gap analyses of various existing knowledge discovery frameworks, evaluating their limitations and practical applications.

5 Conclusions

The UIS-KDF framework is a valuable guide for organizations looking to develop knowledge discovery systems. The separation of technology, management and public applications, constraints, and machine learning layers provide flexibility to adapt to changing business goals. The framework also emphasizes the significance of cybersecurity and the need for meticulous data access and user permissions management. By incorporating these elements, the UIS-KDF framework can help organizations achieve their desired outcomes securely and efficiently.

By providing guidelines for architectural planning and organization, the framework can help organizations develop more effective knowledge discovery systems, leading to improved decision-making.

There are several steps for further research and development related to the UIS-KDF framework, e.g., exploring the challenges of deploying UIS-KDF framework and supporting knowledge discovery system development for companies in specific (vertical) industries.

Acknowledgements Authors have used Grammarly writing assistance platform [38] for grammatics and paraphrasing needs. Chapter content is written by authors exclusively.

References

1. Bindé, J., Matsuura, K., UNESCO (eds.): Towards Knowledge Societies. UNESCO Publications (2005)
2. Sarker, I.H.: Machine learning: algorithms, real-world applications and Research Directions. SN Comput. Sci. **2**(3) (2021). https://doi.org/10.1007/s42979-021-00592-x
3. Jansevskis, M., Osis, K.: Knowledge discovery frameworks and characteristics. Baltic J. Mod. Comput. (BJMC). (2023) [Submitted for publication]
4. Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M.J., Flach, P.: CRISP-DM twenty years later: from data mining processes to data science trajectories. IEEE Trans. Knowl. Data Eng. **33**(8), 3048–3061 (2021)
5. Rotondo, A., Quilligan, F.: Evolution paths for knowledge discovery and data mining process models. SN Comput. Sci. **1**(2), 109 (2020)

6. Cao, L., Zhao, Y., Zhang, H., Luo, D., Zhang, C., Park, E.K.: Flexible Frameworks for Actionable Knowledge Discovery. IEEE Trans. Knowl. Data Eng. **22**(9), 1299–1312 (2010)
7. Jansevskis, M., Osis, K.: User interaction and response-based knowledge discovery framework. Commun. Comput. Inf. Sci. (2023) [Submitted for publication]
8. Wang, J., Yang, Y., Wang, T., Sherrat, R.S., Zhang, J.: Big data service architecture: a survey. J. Internet Technol. **21**(2), 393–405 (2020)
9. Zhu, J.Y., Tang, B., Li, V.O.K.: A five-layer architecture for big data processing and analytics. Int. J. Big Data Intell. **6**(1), 38–49 (2019). https://doi.org/10.1504/ijbdi.2019.097399
10. Karunaratne, P., Karunasekera, S., Harwood, A.: Distributed stream clustering using micro-clusters on Apache Storm. J. Parallel Distrib. Comput. **108**, 74–84 (2017). https://doi.org/10.1016/j.jpdc.2016.06.004
11. Bok, K., Oh, H., Lim, J., Pae, Y., Choi, H., Lee, B., Yoo, J.: An efficient distributed caching for accessing small files in HDFS. Cluster Comput. **20**(4), 3579–3592 (2017). https://doi.org/10.1007/s10586-017-1147
12. Richa, B.: NoSQL vs SQL — Which Database Type is Better for Big Data Applications. https://analyticsindiamag.com/nosql-vs-sql-database-type-better-big-data-applications (2017). Last accessed 13 Mar 2023
13. General Data Protection Regulation (GDPR).: https://gdpr-info.eu/ (2016). Last accessed 8 May 2023
14. Jeren, A.: The impact of the GDPR on Big Data. Tech GDPR. https://techgdpr.com/blog/impact-of-gdpr-on-big-data (2020). Last accessed 9 May 2023
15. Schatz, D., Bashroush, R., Wall, J.: Towards a more representative definition of cyber security. J. Digit. Forensic Secur. Law. (2017). https://doi.org/10.15394/jdfsl.2017.1476
16. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., Wang, C.: Machine learning and deep learning methods for cybersecurity. IEEE Access. **6**, 35365–35381 (2018). https://doi.org/10.1109/ACCESS.2018.2836950
17. Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access. **7**, 41525–41550 (2019). https://doi.org/10.1109/access.2019.2895334
18. Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q.: A deep learning approach to network intrusion detection. IEEE Trans. Emerg. Topics Comput. Intell. **2**(1), 41–50 (2018). https://doi.org/10.1109/tetci.2017.2772792
19. Zhou, L., Pan, S., Wang, J., Vasilakos, A.V.: Machine learning on big data: opportunities and challenges. Neurocomputing. **237**, 350–361 (2017). https://doi.org/10.1016/j.neucom.2017.01.026
20. Technopedia Inc.: Knowledge Discovery. https://www.techopedia.com/definition/25827/knowledge-discovery-in-databases-kdd (2017). Last accessed 10 May 2023
21. Osei-Bryson, K.-M., Barclay, C. (eds.): Knowledge Discovery Process and Methods to Enhance Organizational Performance. CRC Press, Taylor & Francis Group (2015)
22. Ghezzi, C., Jazayeri, M., Mandrioli, D.: Fundamentals of Software Engineering. Prentice Hall (2003)
23. Rowley, J.: The wisdom hierarchy: representations of the DIKW hierarchy. J. Inf. Sci. **33**(2), 163–180 (2007). https://doi.org/10.1177/0165551506070706
24. Yousfi, S., Chiadmi, D., Rhanoui, M.: Smart big data framework for insight discovery. J. King Saud Univ. Comput. Inf. Sci. **34**(10), 9777–9792 (2022). https://doi.org/10.1016/j.jksuci.2021.12.009
25. Rizvi, S., Zwerling, T., Thompson, B., Faiola, S., Campbell, S., Fisanick, S., Hutnick, C.: A modular framework for auditing IOT devices and networks. Comput. Secur. **132**, 103327 (2023). https://doi.org/10.1016/j.cose.2023.103327
26. Khoda Parast, F., Sindhav, C., Nikam, S., Izadi Yekta, H., Kent, K.B., Hakak, S.: Cloud computing security: a survey of service-based models. Comput. Secur. **114**, 102580 (2022). https://doi.org/10.1016/j.cose.2021.102580

27. Zheng, L., Wang, C., Chen, X., Song, Y., Meng, Z., Zhang, R.: Evolutionary machine learning builds smart education big data platform: data-driven higher education. Appl. Soft Comput. **136**, 110114 (2023). https://doi.org/10.1016/j.asoc.2023.110114
28. SAS: Introduction to SEMMA. SAS Help Center (2017). https://documentation.sas.com/doc/en/emref/14.3/n061bzurmej4j3n1jnj8bbjjm1a2.htm
29. Chapman, P., Julian, C., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0: Step-by-step data mining guide. https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf (2000)
30. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: Knowledge discovery and data mining: towards a unifying framework. In: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 82–88 (1996)
31. Rollins, J.B.: Foundational methodology for data science. IBM Analytics. https://tdwi.org/~/media/64511A895D86457E964174EDC5C4C7B1.PDF
32. Severtson, R.B.: What is the Team Data Science Process? https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle (2021)
33. Moyle, S., Jorge, A. (2001). RAMSYS-A methodology for supporting rapid remote collaborative data mining projects.. https://www.researchgate.net/publication/247329752_RAMSYS-A_methodology_for_supporting_rapid_remote_collaborative_data_mining_projects
34. Gokalp, M.O., Kayabay, K., Akyol, M.A., Eren, P.E., Kocyigit, A.: Big data for industry 4.0: a conceptual framework. In: 2016 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 431–434 (2016)
35. Chen, M., Herrera, F., Hwang, K.: Cognitive computing: architecture, technologies and intelligent applications. IEEE Access. **6**, 19774–19783 (2018)
36. Osman, A.M.S.: A novel big data analytics framework for smart cities. Fut. Gener. Comput. Syst. **91**, 620–633 (2019)
37. Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Flach, P., Hernández-Orallo, J., Kull, M., Lachiche, N., Ramírez-Quintana, M.J.: CASP-DM: context aware standard process for data mining. arXiv. https://arxiv.org/abs/1709.09003 (2017)
38. Free writing AI assistance. Grammarly. https://www.grammarly.com/ (n.d.)

Who Guards the Guardians? On Robustness of Deep Neural Networks

Misha Glazunov and Apostolis Zarras

1 Introduction

Deep learning, powered by deep neural networks (DNNs for short), underlies most modern Artificial Intelligence (AI) algorithms. These models have proven highly successful in various applications, including supervised, unsupervised, and reinforcement machine learning. Furthermore, deep learning models have demonstrated state-of-the-art results in processing diverse data domains such as images, texts, audio recordings, electrocardiograms, malware, games, and many more. Despite their impressive accuracy metrics, the fundamental theory of deep learning is missing. First, the most critical lacking aspect is linked to the generalization capabilities of the modern DNNs. Generalization can be subdivided into (i) generalization from training data to test data and (ii) generalization from inliers to outliers. Currently, there is no satisfying theory, even for the first part, which would explain the surprising generalization capabilities of DNNs. Suppose one employs the classical statistical learning theory based on the Vapnik–Chervonenkis dimension [58]. In that case, it likely indicates that DNNs are overfitting, considering that the number of parameters quite often significantly exceeds the number of data points used for DNN training. Many research works contribute to the gradual progress in this direction [4, 12, 35, 50], but no final theory is currently available. The second part is also an active direction of current research [1, 29, 66]; however, nowadays, there is no theoretically rigorous foundation for that either. As a result, both of the parts require further exploration. Moreover, the second missing cornerstone relates to

M. Glazunov (✉)
Delft University of Technology, Delft, Netherlands
e-mail: M.Glazunov@tudelft.nl

A. Zarras
University of Piraeus, Piraeus, Greece

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_6

the optimization routine, e.g., to the convergence regime of DNNs: pointwise vs. uniform. There are several studies in this direction [64, 65, 67] and their connections to the generalization [41], but, again, there is still no rigorous theoretical basis for that. Finally, the architectural choices for DNNs are still empirical, and no theory unifies them except for the very coarse results of the well-known Universal Approximation Theorem (UAT) [10] and its variations that basically aim at rigorous bounding either the depth [3, 22, 49] or the width [36] of the DNNs. The lack of rigorous understanding of the theory of deep learning results in many relevant questions that still need to be answered. For instance, how exactly does the model produce the final result? How can we be sure about these results? What are the limits of a particular modeling approach? How can it be rectified in case of malfunction? And many more others.

Therefore, these models are often used as a black box tool, fed with significant data to achieve the desired performance results. However, this raises serious concerns about the robustness of these models to malicious interventions because if there is no clear understanding of how modern DNNs generalize from training data to test data (i.e., on inliers and, further, on outliers), then it is an obvious direction of exploiting this lack of understanding by probing the models for the potential vulnerabilities, especially when they are used in so many critical domains, ranging from autonomous driving to medical diagnosis. It should be noted that despite the significant progress in the methods of model interpretability such as SHAP [38], GradCAM [54], CAM [69], RISE [48], and CXPlain [53], all of these methods can be applied to any model, and they treat the model in question as a black box with only potential access to the loss function and/or gradients. The interpretations are subsequently derived from input features that indicate the decision made by the model based on these input features and their corresponding contributions or weights for a particular decision. However, none of these methods tell anything about the model's internal mechanics from the fundamental theoretical perspective, e.g., how exactly did the particular DNN come up with this particular solution and why (all the research dedicated to the generalization and optimization of the DNNs)? How does the network architecture influence the final decision and why (vast volume of different ad hoc solutions with too few theoretical foundations)? What can be done to alleviate the wrong output and modify the decision theoretically rigorously (the operational question that logically follows from the first two)? Can we debug the model, identify why these input features are essential for this model, and fix the wrong outputs? (i.e., the practical question based on the previous one).

Consequently, many practical attacks have been discovered that exploit this lack of understanding, which, in turn, assist us a lot in more profound research and comprehension of the internal mechanisms of deep learning. Let us momentarily take the role of an attacker and consider the potentially vulnerable spots that can be exploited from the general perspective in any machine learning modeling approach. Such an approach requires at least two parts: (i) data to be trained on and inferred from and (ii) a particular model to be trained and subsequently used for inference.

Moreover, the attackers may not have access to both model and data. In addition, both the model and data may evolve with time. Furthermore, a model can be

a composite one, i.e., consisting of several other machine learning sub-models, and data can be non-homogenous or multimodal. Based on various combinations of these components, different potential scenarios are available for the attacker depending on their interests. For example, the attackers may be interested in modifying the model's behavior for particular inputs, e.g., they want to bypass a spam filter to organize an advertising campaign for an illegal product. Alternatively, they may want to trick a face recognition system to access the victim's smartphone. Conversely, the attackers may be interested in the data; for example, they may have access to the model and can get inference results but lack knowledge about the dataset used to train the model, which could be of particular interest to them. Alternatively, suppose the attackers cannot access the model. In that case, they may be interested in getting it, e.g., the model architecture, the weights obtained after training, and the hyperparameters used for training.

Therefore, grounded in the distinctive attack vectors, we can discern two principal categories of potential attacks. (*i*) *Functionality-oriented attacks*, encompassing those that seek to manipulate the model's behavior. These may transpire during either the training or inference stages. (*ii*) *Privacy-oriented attacks*, encapsulating those designed to extract insights into the private data employed during both training and inference stages since models may be trained on private data, such as personal information, medical records, or financial data, which can be extracted from the trained model breaching the privacy. Moreover, even if the training data is anonymized, recovering the data the model was trained on may leak proprietary business information. In addition, this attack category may aim to unveil proprietary facets of the machine learning model itself. If the model is deployed in the cloud and used via the web, then the model architecture and parameters leakage represent a serious proprietary trade secrets privacy breach.

The first kind of functionality-oriented attack strongly relates to the unknown inputs, since we assume the model is not under the attacker's control. Hence, something should be done with its input to modify the model behavior. Unknown inputs can be further divided into two main categories. The first category includes inputs that differ significantly from those on which the model has been trained. For instance, if a deep neural network classifier is trained to classify handwritten digits, how will it process images of unknown numerical systems, languages, or even natural objects such as trees or dogs? Experiments demonstrate that DNNs overconfidently fail when processing such inputs as they cannot distinguish between them and those they have been trained on [47, 60]. Such inputs are closely related to the question of the proper model generalization, and they are referred to as outliers. The model deployed in the wild should be able to deal with the outliers, i.e., with the data points coming from a different distribution than the one the model trained on [20]. In such a case, the question arises of how good the model is in dealing with such inputs and, most importantly, if it can distinguish outliers versus inliers. The second category includes inputs specifically manufactured by adversaries to change the model's output, such as adversarial examples. These examples are becoming increasingly prevalent across all learning approaches, model architectures, and data domains.

The attacker utilizes outliers and adversarial examples in the inference stage when the model is already pretrained. Alternatively, the poisonous attacks are introduced during the training stage. These are also artificially forged inputs that tend to be covertly introduced in the training dataset with the aim of the subsequent exploit during the inference stage.

The second kind of privacy-oriented attacks includes model stealing and membership inference attacks.

In this chapter, we briefly introduce the available attacking techniques and indicate the potential directions for their mitigation and, if possible, detection.

2 Attacking Deep Neural Networks

In this section, we introduce appropriate threat models together with potential vectors of attacks at DNNs. It includes both functionality-oriented and privacy-oriented threats. The former comprises outliers, adversarial, and poisonous examples. The latter includes attacks that aim at stealing private or sensitive data related either to the model parameters and architecture or to the inputs used during the training of the model, i.e., to the training dataset. Finally, these attacks are mapped within the confidentiality, integrity, and availability triad components.

2.1 Threat Models

We consider the two most common threat models concerning the attacker's abilities: white box and black box scenarios. The former relates to the situation when the attacker knows, for instance, the DNN model, including its architecture, learned weights, used hyperparameters, input layer representation, and output layer results. This scenario implies that the attacker has full knowledge of both model and training data in use.[1] Conversely, the latter applies to cases without direct knowledge of the DNN model. The attackers may only know the inputs used for the model. Sometimes, they can also use the model as a service via a particular API call to infer the outputs. Moreover, it usually denotes the lack of exact knowledge about the data used during the training stage of the model. Nevertheless, the general understanding of the data is particularly always available or can be inferred, such as, for example, in the case of face detection on the image, the training dataset should have included images with faces.

The privacy-oriented attacks imply the black box scenario. Functionality-oriented attacks, on the other hand, may concern both threat models. The usual approach for them in the case of a black box is to train the so-called surrogate

[1] Please note that it is not in contradiction with our previous assumption about the absence of the control over the model by the attacker since the knowledge of the model still does not imply the access to the model in use.

model, which will be used for the proper adversarial or poisonous input generations and the subsequent evaluations of the attack success rates. The properties of this surrogate model depend on the DNN under attack, and it usually entails the initial reconnaissance step with respect to the model in use potentially involving privacy-oriented attacks.

Beyond the available knowledge of the attackers, as we mentioned before, there is also a distinction between their possibilities over data manipulation, namely, if they can manipulate data during training or inference stages. The first type of attacks is called poisonous or sometimes causative, and the second type is called exploratory or evasion attacks. The latter includes both adversarial examples and outliers. We will use both mentioned terms for each of the types interchangeably.

2.2 Functionality-Oriented Attacks

First, we explore the most active domain of the current research dedicated to misleading the deep learning model behavior utilizing specific inputs. Since such inputs can be provided during the training or inference stage of the DNN life cycle, we dedicate a separate section to each stage.

2.2.1 Exploratory Attacks on Inference

Inference relates to the stage when the model is already trained and validated on a particular dataset. Hence, it can now *infer* the parameters of the probabilities of the general population based on particular provided inputs. Two types of attacks can be associated with the inference stage of any DNN: (i) the attacks utilizing the outliers and (ii) the attacks employing the adversarial examples. In the following sections, we consider each of them in more detail.

2.2.2 Outliers

We begin our discussion with the outliers. We consider that outliers are generated by a different distribution than the inputs used during training. An input generated from the different distribution represents a traditional definition of outliers, and to the best of our knowledge, it was first employed by Hawkins [20].[2] The treatment of these inputs is often considered somewhat parallel to the attacker's perspective in the scientific literature. The reason is that they are more vividly related to the question of the proper model generalization, i.e., to the ability of the model to infer

[2] Hawkins distinguishes two categories of outliers: (i) inliers and outliers adhere to the same distribution and (ii) inliers and outliers generated by different distributions. In this chapter, we use the latter.

the actual parameters of the probability distributions of the general population rather than to the attacking technique. Despite their traditional treatment in the research community, they can also be used to mislead the model behavior by an attacker; the model deployed in the wild should likely be able to deal with the data points coming from a different distribution to the one on which the model has been trained. For a toy example, let us consider the model trained on the MNIST dataset [31]. This dataset contains the handwritten digits. The outliers in such cases could be the inputs from a different dataset, such as FashionMNIST [61] that contains various garments or even randomly generated images. In such a case, the question arises: *How good is the model in dealing with such inputs, and, most importantly, can it distinguish outliers versus inliers?*

To better understand what the outliers mean, it is helpful to pose the following question about a model in plain English: *Does a trained machine learning model know what it does not know?* To answer this question, we first have to understand what it means exactly for the model to deduce if it knows or does not know something. First, recall that DNNs represent universal mapping approximators, i.e., they can learn any function on a compact domain with an arbitrarily close level of precision. However, what type of functions do they learn in most of the applied tasks? *Probability functions!* More precisely, DNNs parameterize either probability density or probability mass functions based on a particular input. This chapter considers attacks on deep learning models based on the supervised discriminative approach, such as DNN classifiers. Hence, such a modeling approach implies parameterizing a conditional distribution over target values y conditioned on the input $\mathbf{x}$: $p_{\boldsymbol{\theta}}(y|\mathbf{x})$. The training of DNNs allows identifying optimum parameters $\boldsymbol{\theta}^*$ based on a stochastic first-order optimization algorithm such as gradient descent. In the case of classification tasks, the standard choice for $p_{\boldsymbol{\theta}}(y|\mathbf{x})$ is a categorical distribution, in the case of regression—a Gaussian distribution (quite often with a constant variance). It may seem only logical to rely on the probability density in answering our question, namely, if the density is low, then the model is uncertain about the input, which can be interpreted as the model not knowing the current input.

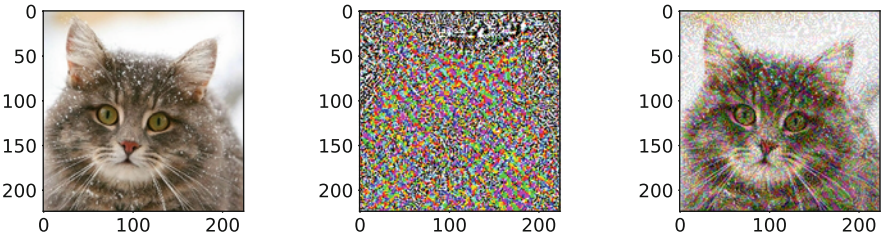
Nevertheless, despite the neat theoretical motivation, such an approach drastically fails in practice. The DNNs tend to assign relatively high probability values to the outliers. For example, consider an ImageNet dataset used in Large Scale Visual Recognition Challenge (ILSVRC) [51]. This dataset contains over a million images for 1000 object classes, including various types of animals, aircrafts, fruits, vegetables, etc. However, despite the vast coverage of potential objects, these 1000 classes do not include images of a microscope or a measuring tape. What would be the probability value for these unknown categories if we train a DNN classifier on ILSVRC based on those 1000 classes? Surprisingly, it turns out to be relatively high! For example, in the case of an image with a microscope, the DNN is 98.18% confident that it is a joystick, and in the case of an image with a measuring tape that it is a chainsaw with a 90.54% probability. Both joystick and chainsaw are present among those 1000 classes, but what is astonishing is the certainty with which the DNN claims to detect objects it has never seen before.

This experiment demonstrates an interesting fact about DNNs: they tend to be *overconfident* with the outliers in their predictions. Such overconfidence has at least two disadvantages. First, it makes it impossible to detect the outliers during the inference stage utilizing the probability values, making such DNNs less suitable for the tasks when such functionality is necessary. Second, it allows an attacker to exploit this overconfidence by allocating particular outliers and providing them as inputs to such DNN, misleading the initially intended behavior, e.g., resulting in misclassification. It is pretty easy to imagine that the impact of such an attack might be life-threatening in many domains, ranging from self-driving cars to medical diagnosis.

2.2.3 Adversarial Examples

Analysis of the input-output mappings of AlexNet Convolutional Neural Network (CNN) [28] from the point of view of continuity reveals unforeseen properties of the DNNs [56]. Namely, the traversal over the manifold learned by the CNN from one category to another with parallel visualizing of the corresponding points in the input space does not result in a smooth morphing of one category to another, e.g., imagine a gradual and substantial transformation of a dog class into a cat class in the resulting images. On the contrary, the results of such a traversal turn out to be quite intriguing; namely, when the CNN classifier indicates that it already observes a different class, the input image still contains a distinct representation of the initial category (e.g., a category of a dog) with a slight amount of the added noise over the image. This means that from the human perspective, the obtained result is almost indistinguishable from the initial image. However, from the perspective of CNN, the new image represents a different category. In parallel, predictably, analysis of the DNNs' robustness against the evasion attacks at test time [5] reveals the same intriguing behavior as in [56]. Two different perspectives on this behavior give rise to two definitions of the adversarial examples: one from the perspective of the generalization properties of the DNN and the other from the attacker's perspective.

From the generalization perspective, an adversarial example [56] is a technique in which the input for the DNN image classier is intentionally modified to look almost the same as the original image to the human eye. Yet, it is perceived as something completely different by DNN. DNNs incorrectly classify such adversarial examples from the human perspective. On the other hand, the attacker perspective does not necessarily demand the part that relates to the imperceptibility of the difference. On the contrary, if the miscreants want their attack's outcome to succeed, they should not constrain themselves to the superfluous imperceptibility demands. In this chapter, nevertheless, we are more inclined to the definition that involves the imperceptibility aspect. The reason for that is the new, often overlooked, category we include in our study: the outliers. These inputs could be considered similar to the perspective of the *adversarial attack to the maximum extent* when there is nothing similar between the original data and the outliers. In particular, we consider three



(a) Persian cat (b) Constrained gradient of the loss (c) Peacock ($\varepsilon = 0.25$)

Fig. 1 (**a**) the original image; (**b**) the max-norm constrained gradient of the loss generated by FGSM; (**c**) the resulting adversarial image

methods of generating adversarial examples: (i) fast gradient sign method [17], (ii) Carlini-Wagner attack [7], and (iii) Jacobian-based Saliency Map attack [46].

Fast Gradient Sign Method The fast gradient sign method (FGSM) attacks DNNs by leveraging their learning process based on gradients [17]. The following formula describes the process of generating an FGSM example:

$$\mathbf{x}' = \mathbf{x} + \epsilon \cdot \text{sgn}(\nabla_{\mathbf{x}} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}), y_s)) \tag{1}$$

Here $\nabla_{\mathbf{x}} \mathcal{L}$ is the gradient of the loss function with respect to the original input pixel vector $\mathbf{x}$, y_s is the actual or source label for $\mathbf{x}$, and $\boldsymbol{\theta}$ stands for the parameters of the model $f_{\boldsymbol{\theta}}$ that are constant. Gradient w.r.t. $\mathbf{x}$ is easier to calculate with backpropagation than for $\boldsymbol{\theta}$, which allows the fast generation of adversarial examples. FGSM exploits gradient ascent to increase the loss. Subsequently, the sign applies a max-norm constraint on the gradient value, and ϵ represents a small magnitude of the step to increase the loss. This formulation constitutes the untargeted type of adversarial attacks, a particular example depicted in Fig. 1.

The FGSM can be converted into a targeted attack by substituting the source label with a target one y_t and doing gradient descent instead of ascent, namely:

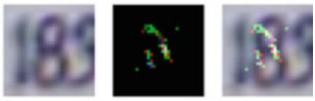
$$\mathbf{x}' = \mathbf{x} - \epsilon \cdot \text{sgn}(\nabla_{\mathbf{x}} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}), y_t)) \tag{2}$$

However, since FGSM is designed to be fast rather than optimal, it is not necessarily guaranteed to produce the targeted adversarial examples of minimal perturbations.

Carlini-Wagner The Carlini-Wagner (CW) attack [7] aims at optimality in contrast to FGSM, i.e., it attempts to generate as little noise as possible to succeed in the attack. It poses the following optimization objective:

$$\text{minimize } ||\boldsymbol{\delta}||_p \quad \text{subj. to } f_{\boldsymbol{\theta}}(\mathbf{x} + \boldsymbol{\delta}) = y_t, \quad \mathbf{x} + \boldsymbol{\delta} \in [0, 1]^n \tag{3}$$

(a) CW applied to the MNIST dataset.



(b) JSMA applied to the SVHN dataset

Fig. 2 (**a**) *Top row:* MNIST adversarial examples targeted for 0. *Bottom row:* MNIST adversarial examples targeted for 1. (**b**) JSMA adversarial features visualized

where $\mathbf{x} \in [0, 1]^n$ represents an image, $\boldsymbol{\delta} \in [0, 1]^n$ is the added noise to the image, and $f_{\boldsymbol{\theta}}$ is a model that returns a target class label of the image under attack. The noise level is calculated in terms of L_p norms. The authors consider several norms; our example demonstrates the L_2-norm. The CW attack represents a targeted attack with powerful properties. Till the moment, it is one of the strongest known adversarial attacks. The examples of the CW-targeted attack on MNIST digits can be observed in Fig. 2.

Jacobian-Based Saliency Map Attack The Jacobian-based Saliency Map Attack (JSMA) [46] leverages the saliency maps to devise an adversarial input. Namely, it computes the forward derivative of the whole DNN (Jacobian) w.r.t. the input, and based on this derivative, it constructs the saliency map. Large absolute values of the saliency map reveal the features significantly impacting the final output. The JSMA takes the maximum absolute value, perturbs it by a hyperparameter θ, and repeats the process. The stopping criteria are either a successful attack with misclassification or reaching the total perturbation threshold of Υ. Figure 2 depicts such a JSMA attack.

2.3 Causative Attacks on Training

This type of attacks implies that the attacker can manipulate some part of the dataset to be subsequently used for training the DNN model under attack. Since this manipulation is always intended as malicious, the term *poisonous* describes this fact rather appropriately and precisely. Generally speaking, there are two possible malicious intents: they aim at intrusion without corrupting the system behavior with respect to benign inputs or target an overall regular system operation, causing a denial of service. First, we consider the way the training data is modified.

2.3.1 Poisoning Dataset

Construction of a poisonous dataset includes collecting raw samples of the statistical population distribution of interest with their subsequent labeling. Depending on

the attackers' capabilities, they may be able to modify raw samples or only their corresponding labels (or both). The first attack is called a clean-label attack, and the second is a dirty or corrupted-label attack. The clean-label attack may seem impossible at first, since there is no control over the output category, and it is unclear how the attacker can exploit the training process in such a case. Nevertheless, two known attacking techniques allow us to achieve this goal. Both introduce slight disturbances to the inputs that mislead DNNs during training. These disturbances are identified through the solution of either a bi-level optimization problem or a feature-collision problem.

2.3.2 Data Poisoning to Disrupt Overall Performance

This type incorporates several indiscriminate attacks, i.e., attacks that do not target a specific category or class within the training dataset. First, we describe a corrupted-label attack based on label flipping. After that, we consider a clean-label attack based on bi-level optimization.

Label Flipping Attack The attacker can significantly reduce the resulting DNN performance only via tampering with the labels of the dataset. It means that it is optional to have access to the model and the raw data. This type of attack can be implemented by mislabeling the data. Labels can be assigned randomly, influencing the system's overall performance; the specific classification category or even several categories can be targeted. The classifier is subsequently trained on the tampered dataset without knowing which labels have been corrupted. The success rate of this type of attack heavily depends on the ratio of the poisoned dataset: the greater the ratio in favor of poisonous examples, the stronger the impact it would have on misclassification.

Attacks via Bi-level Optimization This type aims to solve the corresponding bi-level optimization problem and represents clean-label attacks. Consider the following scenario: attackers want to attack a particular DNN model that will be trained, validated, and tested on the dataset $\mathcal{D}$. They want to poison this dataset to change the resulting model behavior. For that reason, the attackers split this dataset into training and validating subsets $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{val}}$. Subsequently, they aim at poisoning the training dataset $\mathcal{D}_{\text{tr}}$ with a specially crafted input $\mathbf{x}_p$. The crafting of the poisonous input is constrained within the space of allowed manipulations imposed by Φ, i.e., $\mathbf{x}'_p \in \Phi(\mathbf{x}_p)$. As a result, they get a poisoned trained dataset $\mathcal{D}_p = \mathcal{D}_{\text{tr}} \cup \{\mathbf{x}'_p\}$. The goal is to optimize for the optimal poisonous solution allowed within the imposed constraints such that the target input $\mathbf{x}_t$ with the original label y_t from the untampered validation dataset $(\mathbf{x}_t, y_t) \in \mathcal{D}_{\text{val}}$ is misclassified by the model trained on the poisoned dataset.

To achieve this goal, the optimization procedure should be run across two levels. First, the model should be optimized with respect to the DNN parameters on the dataset with the injected poisonous examples, which represents a classical optimization objective of any DNN training, usually referred to as a training loss $\mathcal{L}_{\text{tr}}$. Second,

on top of the classical optimization objective, there is an additional adversarial loss $\mathcal{L}_{\text{adv}}$ that should be optimized for the best poisonous input modification in such a way that this adversarial loss is maximized which eventually leads to the sought-after misclassification. Note that the misclassification is untargeted in this particular case. Hence, this formulation of the optimization objectives represents an attack on the availability of the model aiming at disrupting the classification results for the poisonous inputs.

Formally, both of the optimization levels can be formulated as follows:

$$\mathbf{x}_p^* \in \underset{\mathbf{x}'_p \in \Phi(\mathbf{x}_p)}{\arg\max}\, \mathcal{L}_{\text{adv}}(\mathcal{D}_{\text{val}}, \boldsymbol{\theta}^*(\mathbf{x}'_p)) \tag{4}$$

$$\text{s.t.} \qquad \boldsymbol{\theta}^* \in \underset{\boldsymbol{\theta}}{\arg\min}\, \mathbb{E}_{(\mathbf{x}'_p, y) \sim \mathcal{D}_p}[\mathcal{L}_{\text{tr}}(f_{\boldsymbol{\theta}}(\mathbf{x}'_p), y))] \tag{5}$$

where $\mathbf{x}'_p \in \mathcal{D}_p$, i.e., it is an input from the poisoned training dataset under constrained manipulations allowed by Φ, and $\mathbf{x}_p^*$ is the resulting poisonous input to be added in the original clean dataset.

The process of optimization of this bi-level problem seems straightforward at first glance; the inner minimization procedure obtains the optimal parameters of the DNN with the subsequent maximization of the adversarial loss. This process can be repeated until the convergence. However, in the case of DNNs, this solution is not feasible due to the over-parameterization and impossibility of the exact solution of the inner problem. For that reason, the truncated back-gradient optimization is used [16]. The idea is to first optimize the inner problem for a limited number of iterations, which allows computing the necessary gradients with subsequent backpropagation for the outer objective by tracing the necessary gradients backward. This approach allows the generating of poisonous inputs for DNNs successfully.

2.3.3 Targeted Poisoning Attacks

We consider one corrupted-label targeted attack based on a bi-level optimization objective and one clean-label targeted attack utilizing feature collision.

Attacks via Bi-level Optimization The indiscriminate bi-level optimization poisonous attack described above can be easily changed into a targeted one, if the outer objective is minimized instead, namely:

$$\mathbf{x}_p^* \in \underset{\mathbf{x}'_p \in \Phi(\mathbf{x}_p)}{\arg\min}\, \mathcal{L}_{\text{adv}}(\mathcal{D}'_{\text{val}}, \boldsymbol{\theta}^*(\mathbf{x}'_p)) \tag{6}$$

The dataset for validation contains the same raw inputs as $\mathcal{D}_{\text{val}}$ but with the modified labels that the attacker wants to target, hence minimizing the loss toward them.

Attacks on Feature Collision The intuitive idea behind this type of attacks is to look for the neighbors in the feature space within a relatively close distance from the target class that simultaneously lie close in the input space to the base class, i.e., to the class under attack:

$$\mathbf{x}_p^* \in \arg\min_{\mathbf{x}} \ \| f_{\boldsymbol{\theta}}(\mathbf{x}) - f_{\boldsymbol{\theta}}(\mathbf{t}) \|_2^2 + \beta \, \| \mathbf{x} - \mathbf{b} \|_2^2 \tag{7}$$

Given the feature space's usual complexity and high dimensionality, such a search appears feasible. Note that due to the closeness to the target class in the feature space, there is no need to modify the label, making this attack clean-label. Moreover, the closeness of the poisonous sample to the base one in the input space does not raise concerns during the visual inspection, since the label is correctly assigned as if it represents a base class. Both of these factors make this attack very powerful.

2.3.4 Data Poisoning for Intrusion

In this category, the attacker introduces a set of poisonous inputs that contains a specifically crafted trigger that serves as a backdoor, allowing the intruder to modify the behavior of the DNN in the desired direction [9].

Backdoor Patches The first attack from this category exploits virtual or physical visible patches [18]. The training dataset is poisoned with the patched instances. Subsequently, the DNN is trained with the *corrupted-labels* for the samples that contain the corresponding patches. These patches serve as a trigger for the DNN classifier, allowing the attacker to change the classification result of the model. Since the training dataset contains benign and patched inputs, the normal behavior of the model with benign inputs and classification test errors are not influenced by these patched samples. Examples may represent a patch on the traffic stop sign to misclassify it as a traffic sign for the speed limit (see Fig. 3a). Please note that such a patch can be imposed with the original image during the construction of

(a) Patch (b) Semantical (c) Blending (d) Signal (e) Imperceptible

Fig. 3 Different types of backdoor attacks: (**a**) artificial patch with a corrupted label, (**b**) benign semantical trigger with a corrupted label, (**c**) blending a fake reflection with a clean label, (**d**) introducing a strong signal with a clean label, and (**e**) backdoor imperceptible to the human eye with a clean label

the poisonous input without needing a physical photo with the patch. However, sometimes it is reasonable to rely on some physical trigger that carries a distinct semantical meaning but looks benign in contrast to the patching, for example, when a person wears glasses (see Fig. 3b).

Backdoor as a Strong Signal Several functional approaches could be applied to implement a *clean-labels* backdoor. First, it is possible to bind a strong signal function that any DNN can easily detect with a particular target class, e.g., a sinusoidal signal with a ramp can be mixed into the small fraction of the target images with subsequent use of this signal as a backdoor for any other image during the inference stage. The sufficient ratio of poisonous examples before training the model for the successful attack constitutes one-third of the target class for the MNIST dataset and one-fifth of the traffic signs dataset [2] (see Fig. 3d). The problem with the sinusoidal ramp is that it is still quite visible in the resulting image, which can be detected via visual inspection of the dataset.

Another type of signal can be stealthily encapsulated into the target image representing a fake reflection [33]. The benefits of this approach are that it cannot be easily detected as the previous signal and taking into consideration that it is also a clean-label poisonous attack then it provides all the benefits of a nice backdoor in cases when the target images have reflecting surfaces (see Fig. 3c).

Imperceptible Backdoor Previous backdooring techniques still require a visible trigger and a relatively high ratio of poisonous examples to be added to the training set. The ultimate backdoor, however, can be forged utilizing the technique that we have already described above, namely, feature collision [55]. Such an approach allows the injection of a backdoor even by poisoning one image only. Moreover, thanks to the optimizing within the deep feature representation while minimizing the difference in the input domain, the resulting poisoned image contains imperceptible perturbances to the human eye, making this backdoor one of the most potent poisoning attacks (see Fig. 3e).

2.4 Transferability

It has been discovered that different architectures of DNNs trained to tackle the same classification problem on similar datasets tend to have similar fairly piece-wise linear decision boundaries that separate categories in the input data domain [17]. This property is called transferability. Transferability is especially dangerous, since it allows devising either an adversarial example or poisonous input that universally targets all DNNs with a similar final objective in a black-box manner [45].

The thorough study of the transferability properties indicates the difference between adversarial and poisonous samples. In particular, the adversarial examples tend to transfer better when forged on a simplified surrogate model. However, for poisonous attacks, the best surrogate models are the ones that match the complexity of the target [11]. Moreover, in the case of the white-box threat model, both

adversarial and poisonous inputs favor the more complex models, i.e., the success rate of the attacks is higher for a more complex model than for a simpler one.

2.5 *Privacy-Oriented Attacks*

These attacks aim at stealing any private data. The first type of privacy-oriented attacks involves stealing the pretrained model's proprietary parameters, including weights and hyperparameters. The model in this scenario is usually queried remotely via the web. The second type relates to identifying if a particular input was in the training dataset, allowing the attacker to deduce the data on which the model under attack was trained. In this section, we describe both of these types.

2.5.1 Model Stealing

Nowadays, the functionality provided by various DNNs is often offered online as a service. Such practice amplifies the significance of securing the privacy-related information related to the models deployed in the cloud since its leakage may induce confidentiality-related infringements and substantial financial costs.

Attacks that aim at model stealing attempt to extract this information somehow and, eventually, to reconstruct the existing target model. It may concern the reconstruction of the entire model along with its weights, the reconstruction of the hyperparameters, or the reconstruction of a functionally equivalent alternative model. The adversary gathers the data about the target model by sending a data sample and receiving the model's prediction. This interaction is termed *query-based*.

Equation-solving Attacks Equation-solving attack (ESA) is based on formulating and solving a system of equations, the solution of which yields desired values for an adversary. It aims at specific values of the target model, e.g., learned parameters or training hyperparameters. Using model outputs $y_1, ..., y_n$ for given data samples $\mathbf{x_1}, ..., \mathbf{x_n}$, it's possible to construct equations $f_{\boldsymbol{\theta}}(\mathbf{x_i}) = y_i, i = 1, ..., n$, revealing parameter values $\boldsymbol{\theta}$ [57]. ESA is quite efficient: depending on the target model type, 1 to 4 queries per parameter is enough. This attack requires the knowledge of the target model's architecture and the data samples to query the model. For black-box access, training hyperparameters need prior architecture and parameter extraction attacks.

Meta-model Training Attacks The meta-model attack (MMA) is the only query-based attack capable of uncovering target model architecture to date. A meta-model is trained using a set of candidate CNNs with varying architecture, optimization, and data parameters as the meta-model's dataset, predicting a model's structure, training setup, and data volume. The meta-model then links model hyperparameters and performance through specific test samples, revealing target model hyperparameters [42].

MMA's success depends on the hyperparameter's influence and their presence in the training set. For instance, to steal the convolutional layer count, the target model must be a convolutional neural network and possess a corresponding number of layers in the training set for the meta-model. Execution demands considerable computational resources: for MNIST classifiers, 10,000 CNN candidates had to be trained for over 40 days on a GPU. On average, the attack correctly predicted hyperparameters 80.1% of the time, surpassing a 34.9% guessing chance. However, as MMA extracts hyperparameters, an extra parameter-stealing attack is necessary to approximate the whole target model behavior [43].

2.5.2 Membership Inference

Membership inference involves the identification of training data associated with a trained model. When provided with a data instance and granted access to a model, the objective is to detect whether this data instance was a part of the model's training dataset. The access to the model can vary, falling into either the white box or black box category.

Membership inference attacks stem from the fact that a machine learning model might behave differently on the training dataset than the test dataset. It is particularly evident in machine learning, especially within DNNs, which often are over-parameterized when the number of trainable parameters exceeds the number of training instances. It enables a machine learning model to "remember" instances from the training data. Thus, it may assign significantly higher confidence to predictions for training instances than test instances. By exploiting such differences, attackers can deduce whether a given instance is likely to belong to the training dataset.

2.6 CIA Triad

The classical information security triad comprises three main components: confidentiality, integrity, and availability (CIA). We can now look at the described attacks through the CIA's prism. The functionality-oriented attacks relate to both integrity and availability. The integrity component involves the attacks that maximize the model Type-I errors, i.e., adversarial examples and backdoor attacks. Availability attacks aim at minimizing the utility of the DNN model, e.g., by increasing the model Type II errors via poisonous attacks or by flooding the model with outliers, exploiting model overconfidence, and making the model predictions irrelevant in most cases. The privacy-oriented attacks, however, are connected with the confidentiality component. It includes both model stealing and membership inference. The summary of all of these aspects, including the attacker's capabilities, can be observed in Table 1.

Table 1 Summary of the attacks and links to the CIA triad. Enhanced from Biggio and Roli [6]

Attacker's capability	Attacker's goal		
	Functionality-oriented		Privacy-oriented
	Integrity	Availability	Confidentiality
Train data	Poisoning for subsequent intrusion	Poisoning to invalidate model	–
Test data	Adversarial examples	Outliers	Model stealing, membership inference

3 Available Defenses

This section covers several available defense strategies against the attacks mentioned above.

3.1 Defense Against Functionality-Oriented Attacks

As for the available defenses against functionality-oriented attacks, it should be stressed that the current state of the art represents a constant race between the invention of new defenses against known attacks and the subsequent breaking of this defense with the newly discovered vulnerabilities. Nevertheless, in this section, we introduce the defense methods that demonstrate promising robustness results, and many of them have already passed the test over time.

3.1.1 Adversarial Training

Currently, there is no readily available universal defense mechanism that provides complete protection against adversarial examples. Nevertheless, several techniques proved promising in mitigating the potential damage and consequences, one of them being adversarial training [17, 56]. The transferability not only allows a black box adversarial attack but also means that it is possible to generate the known adversarial examples in advance automatically and add them to the training set before starting the training. Such an approach forces the DNN model to consider the adversarial perturbations and increases the robustness of DNNs to adversarial attacks.

The advantage of this method is its simplicity. The most significant disadvantage is the requirement to generate adversarial examples in advance, which may defend only against attacks known during the DNN model training. Since adversarial examples can be generated with a vast diversity of different techniques, this approach can be helpful only if the model retraining and redeployment procedures

are relatively cheap. Thus, keeping the model up-to-date with developing new attacks is possible.

3.1.2 Enhanced Optimization Objective for Adversarial Training

The idea of this approach is first to enhance the standard optimization objective and then exploit one of the adversarial attacking techniques for approximating one of the stages of the optimizations. Namely, recall the standard loss objective for the classification task:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}), y))] \tag{8}$$

where $\mathbf{x}$ is the input, y is the corresponding label, $\mathcal{D}$ is the training dataset, and $\boldsymbol{\theta}^*$ are the optimal trained parameters of the DNN.

This objective may be improved, so that its exact solution would *guarantee* the robustness against the adversarial examples [39]. Specifically, it can be achieved by requiring that every neighbor of the current point within the ϵ-ball adheres to the same class:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\max_{\|\mathbf{x}'-\mathbf{x}\|_p\leq\epsilon} \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}'), y))\right] \tag{9}$$

Unfortunately, the exact solution for optimizing this min-max objective is unattainable in a reasonable amount of time, since it involves a standard process of first a non-concave inner maximization and then a non-convex outer minimization. The solution is to approximate the worst-case inner approximation problem by optimizing for adversarial examples. In particular, FGSM (see 1) can be interpreted as a single step for maximizing the inner part of this objective formulation. In order to get a more precise approximation to the solution, one can also apply a multistep attack such as projected gradient descent (PGD) [7].

3.1.3 Outlier Exposure

Similar to the idea of adversarial training is the outlier exposure approach [21], introduced for DNNs to deal with the outliers. The idea is to enhance the training dataset of the inliers $\mathcal{D}_{\text{in}}$ with the outliers. Since the dataset with the real outliers $\mathcal{D}_{\text{out}}$ is not available, it is though possible to enhance the dataset with the different available datasets for Outlier Exposure $\mathcal{D}_{\text{out}}^{\text{OE}}$. This dataset is different from the inliers. The model f is subsequently trained to learn more conservative signals of the inliers versus provided outliers that enhance robustness against previously unobserved outliers. It is achieved by introducing an additional term to the loss function, which is responsible for the outlier detector:

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{in}}}[\mathcal{L}(f(\mathbf{x}),y)+\lambda\mathbb{E}_{\mathbf{x}'\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[\mathcal{L}_{\text{OE}}(f(\mathbf{x}'),y)]] \tag{10}$$

The first term of the loss is the standard cross-entropy loss $\mathcal{L}$ used in the classification tasks where $\mathbf{x}$ represents the input and y stands for the corresponding label. The second term includes the outlier exposure loss $\mathcal{L}_{\text{OE}}$ for the outlier detector. It represents the cross-entropy from $f(\mathbf{x}')$ to the uniform distribution. Unlike the adversarial training approaches described in the previous sections, this method allows us to mitigate and detect the outlier.

3.1.4 Defense via Generative Models

Several defenses based on the Deep Generative Models (DGMs) do not imply any knowledge about the type of adversarial attack in use. Moreover, they can be applied independently of the classifier DNN, since they are used as a filter working with the input before passing it further to the DNN model under protection. In addition, some DGMs allow the detection of outliers in a completely unsupervised manner that can be combined with the adversarial filtering approach, enhancing the robustness against outliers and adversarial examples.

DGMs as Adversarial Filters This approach exploits the DGMs' ability to learn a joint distribution of the data that results in a different learned representation compared to the discriminative approach, e.g., in the case of a DNN classifier. The most obvious difference with the discriminative representation is that the generative representation allows the generation of new samples from the learned joint distribution that look similar to those observed in the dataset during the training stage. Based on this representative power, it becomes possible to train a separate DGM model that will work as a filter, i.e., any input is first fed to the DGM filter, which mitigates the ongoing or attempted attack [23, 52]. The mitigation requires normal DGM training on the same dataset as the DNN under protection. As a result, it allows projecting the adversarial input onto the range of the DGMs' generator based on the learned representation by minimizing the reconstruction error. Hence, by default, this method includes filtering out the adversarial perturbations. Finally, the protected DNN classifier will get the clean input without adversarial perturbations.

DGMs as Unsupervised Outlier Detectors The methods within this category allow the detection of outliers completely unsupervised. They can rely either on the ensemble-based epistemic uncertainty estimation [14] or the DGM latent representation [15]. In both cases, the DGM can be again used as a filter to the DNN under protection. However, this time, it is not only mitigating the attack but also detecting if the current input falls under the category of outlier or not. Moreover, it turns out that the outliers tend to gravitate toward the holes in the latent representation of DGMs (see Fig. 4), specifically in the variational autoencoders (VAEs), allowing its simple detection. Finally, the same DGM that is used for the

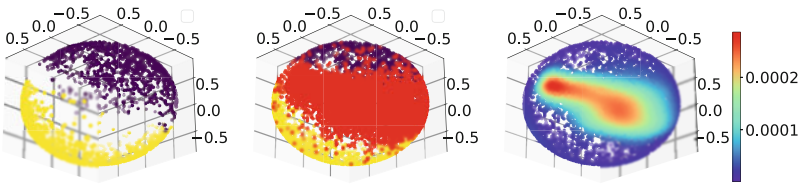


Fig. 4 Compact spherical latent space of VAE trained only on two digits of the MNIST dataset: 0's and 1's. It is a vivid demonstration of the fact that the outliers densely land on the hole in the latent representation. *From left to right*: Yellow depicts means of the estimated posteriors for 1s and purple for 0s; red represents the mapped means for a held-out outlier: a class of digits 9s; Kernel density estimation of all means with the densest region in the hole packed with the outliers

outlier detection can be applied in the same logic as the previously described DGMs for the adversarial mitigation, i.e., if the outlier is not detected, then the input is purified from the adversarial perturbations and subsequently fed to the DNN under protection. This approach is auspicious since it targets adversarial examples and outliers utilizing only one filtering model.

3.1.5 Defense Against Poisonous Attacks

Defenses against poisonous attacks can be broadly categorized into one of the following groups:

1. *Data Level Defenses:* Focus on removing poisonous data from the input.
2. *Model Level:* Involve model retraining.
3. *Interaction between Data and Model Levels:* Identify patterns in the interaction between the data and model training.

Data Level Defenses The first approach to the defense in this category is treating the poisonous samples in the tampered dataset as outliers. It implies that the defenders can access the benign dataset to train an outlier detection model. For example, they can train another model on the benign dataset with a different architecture from the original one but with comparable accuracy metrics [30]. Subsequently, any new input can be forwarded through both models, and if there is disagreement in the predictions between them, then this input is marked as potentially poisonous.

Another approach is based on the intentional input perturbation when the defender modifies the input by blending it with another benign input. It has been noticed that the poisonous backdoor inputs are still successfully classified as the target of the backdoor, whereas the blending of two benign inputs produces a random prediction result [13]. This difference is most likely due to the special optimization procedure used when forging a backdoor input, in which the objective is set so that it should overcome the original trigger "blending," resulting in a successful attack even when additional blending is applied.

Model Level Defenses The defenses of this type imply that the defender has access at least to the smaller subset of the original benign dataset. In such a case, the defender can run a training procedure based on fine-tuning the poisoned model utilizing this benign subset. The benign data can be simultaneously augmented with the samples by adding random Gaussian noise to make the defended model even more robust to potential poisonous input perturbations. After several iterations of such fine-tuning, the model becomes robust and immune to the previously poisonous inputs [34, 59].

An alternative approach within this type of defense is based on meta-classification when an additional DNN is trained on the features extracted from the defended DNN to classify if this model has been compromised by poisoning. The defender has to construct a set of DNNs evenly divided into poisoned and benign ones. After that, the meta-classifier is trained based on the features extracted from these DNNs with a single purpose to distinguish between the benign and tampered models. This approach demonstrated good generalization results even with the before unseen poisonous techniques [27, 63].

Defenses Based on the Interaction between Data and Model Training This type of defense is relatively novel and looks like the most robust and promising one. It does not imply that the defender has access to the benign dataset for outlier detection or a benign subset of the initial dataset for subsequent model retraining using fine-tuning. It does not rely on the meta-classification either, implying that the defender has access to benign and poisonous models. Instead, this defense is working its way out utilizing the provided dataset and the given training objective interchangeably, resulting in the clustering of the samples based on the incompatibility property [25]. The intuition is that the benign inputs should improve the optimization objective during the training or at least not degrade it; however, the poisonous inputs, on the contrary, should reduce the validation accuracy. This idea leads to the clustering method that iteratively builds up clusters based on the incompatibility with respect to the training objective, i.e., benign inputs will eventually generalize to themselves as opposed to the poisonous inputs that would be clustered separately in such cases.

3.2 Defense Against Privacy-Oriented Attacks

In this section, we cover both mitigation and detection defensive methods against privacy-oriented attacks. The mitigation methods focus on minimizing the attack's impact, i.e., they do not stop the attacker from acquiring a model; their goal is to reduce the stolen model's quality to a point where it becomes unusable. The detection methods can further be subdivided into ownership verification (usually achieved by unique model identifiers or watermarking that can prove ownership of a stolen model; aims at proving past attacks) and attack detection (monitors whether a model is currently being attacked). Attack detection cannot prevent a model from being stolen, but it can inform the owner about the incident.

3.2.1 Basic Defenses

We begin with a listing of several simple basic defenses that can be applied to mitigate attacks that aim at stealing the DNN model. These defenses demonstrate good protection results despite their simplicity.

Input Modification Defenses The first type involves input modification, i.e., a defender can modify the input provided to the DNN so that only insignificant parts of this input are modified. For example, in the case of CNNs and image classification tasks, these parts would represent insignificant pixels for the resulting classification. If the defender adds random noise to these pixels, it will make it very complicated for the attacker to steal the DNN's parameters [19]. The technique that identifies such insignificant pixels in the first place is based on the Gradient-based CAM [54] that was initially developed for the visual interpretability of the decisions made by CNN in the input feature space.

Ouput Modification Defenses The second type implies output modification. It is based on rounding of the predicted values [57]. This rounding prevents the attacker from stealing the model's parameters due to the impossibility of solving the corresponding system of equations. Another alternative is to utilize the so-called adaptive misinformation [26]. It is a special technique to return wrong predictions for user queries but in an adaptive way. The intuition behind this approach is to notice that model attackers quite frequently use queries that lie out of the distribution. Based on this observation, it is logical to adapt a model so that it will assign wrong predictions to such queries, making it much more complicated for the attacker to steal the model.

Model Modification Defenses In addition to the perturbation of inputs and outputs, modifying the model architecture and parameters is possible. A CNN feature extractor can be simulated to train a simpler model, namely, a shallow and sequential convolutional block, via Knowledge Distillation (KD) [62]. Such an approach is akin to the source code obfuscation technique obstructing the attacker from stealing the initial proprietary model parameters and architecture. Moreover, it is possible to choose an opposite way by adding redundant layers that do not change the functionality, which makes theft of the model much more complicated [8].

3.2.2 Unique Identifiers and Watermarking

This section addresses several ownership verification defensive techniques.

Unique Model Identifier Unique model identifier (UMI) is a detection method that identifies a distinctive model property that transfers to a substitute model during theft. By revealing this property, a model owner can prove the model was stolen. This technique does not require an active embedding of this unique, distinctive property in contrast to watermarking, as it is inherent to the model. One example of this technique is a dataset inference (DI) defense. DI detects if a model was

trained on a specific dataset [40]. This method is based on the idea that training samples lie farther from the decision boundary than other samples. A subset of the original training data measures the distance of samples from the boundary in the substitute model. If they are distant, the substitute model contains the target model's identifier, indicating a potential model theft. However, DI is ineffective when the original dataset is public, misclassifying independent models trained on it as stolen [32]. Alternatively, conferrable adversarial examples can form unique fingerprints for substitute models [37]. These samples represent a unique set of adversarial examples that transfer to the substitute models but not to the other independently trained models.

Model Watermarking Another approach is based on the active intervention in the model training process so that the owner changes the model behavior on some or all inputs so that only the owner can identify these changes. Model watermarking (WM) actively embeds concealed data to establish ownership. Unlike UMIs, it involves secret backdoors that make models predict predefined values for some data containing outliers, revealing watermarks. However, embedded watermarks also have to be persistent to model stealing, i.e., they have to be identifiable in the model after it has been stolen. This property can be achieved by training a model that extracts common features from problem-domain inputs and watermarking samples. Such an approach ensures that watermarks would also be extracted during the model theft [24].

3.2.3 Monitor-Based Defenses

Monitor-based detection defense involves query analysis to identify malicious users. For example, a defender can train a dedicated discriminative DNN to classify adversarial versus benign samples, using each hidden layer outputs of a protected DNN as features [68]. Alternatively, a DGM such as VAE can also differentiate benign from malicious queries [44].

4 Conclusion

In this chapter, we have covered the broad range of attacks on discriminative DNNs. We have touched upon all potentially vulnerable spots, including the influence on the model's behavior and the potential confidential data leakage. In addition, we detailed the most prominent examples of adversarial and poisonous attacks, including the potential adversarial usage of the outliers and various backdoor techniques. Furthermore, we delved into two powerful privacy-related attacks: model stealing and membership inference. Lastly, we highlighted the most promising and robust defense mechanisms that are currently available in the arsenal of the DNN developer to either mitigate or detect the undergoing attack.

References

1. Ahuja, K., Caballero, E., Zhang, D., Gagnon-Audet, J.C., Bengio, Y., Mitliagkas, I., Rish, I.: Invariance principle meets information bottleneck for out-of-distribution generalization. In: Advances in Neural Information Processing Systems (NIPS) (2021)
2. Barni, M., Kallas, K., Tondi, B.: New backdoor attack in CNNs by training set corruption without label poisoning. In: IEEE International Conference on Image Processing (ICIP) (2019)
3. Barron, A.R.: Approximation and estimation bounds for artificial neural networks. Mach. Learn. **14**(1), 115–133 (1994)
4. Belkin, M., Hsu, D., Ma, S., Mandal, S.: Reconciling modern machine-learning practice and the classical bias–variance trade-off. Proc. Natl. Acad. Sci. **116**(32), 15849–15854 (2019)
5. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In: Machine Learning and Knowledge Discovery in Databases (2013)
6. Biggio, B., Roli, F.: Wild patterns: ten years after the rise of adversarial machine learning. Pattern Recogn. **84** (2018)
7. Carlini, N., Wagner, D.: Towards Evaluating the Robustness of Neural Networks (2016)
8. Chabanne, H., Despiegel, V., Guiga, L.: A Protection against the Extraction of Neural Network Models (2020)
9. Cinà, A.E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B.A., Oprea, A., Biggio, B., Pelillo, M., Roli, F.: Wild patterns reloaded: a survey of machine learning security against training data poisoning. ACM Comput. Surv. **55**(13s) (2023). https://doi.org/10.1145/3585385
10. Cybenko, G.: Approximation by superpositions of a sigmoidal function. Math. Control Signals Syst. **2**(4), 303–314 (1989)
11. Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C., Roli, F.: Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In: USENIX Security Symposium (2019)
12. Dziugaite, G.K.: Revisiting generalization for deep learning: PAC-Bayes, flat minima, and generative models. Ph.D. Thesis, University of Cambridge (2020)
13. Gao, Y., Xu, C., Wang, D., Chen, S., Ranasinghe, D.C., Nepal, S.: Strip: a defence against trojan attacks on deep neural networks. In: Annual Computer Security Applications Conference (ACSAC) (2019)
14. Glazunov, M., Zarras, A.: Do Bayesian variational autoencoders know what they don't know? In: Conference on Uncertainty in Artificial Intelligence (UAI) (2022)
15. Glazunov, M., Zarras, A.: Vacant holes for unsupervised detection of the outliers in compact latent representation. In: Uncertainty in Artificial Intelligence (UAI) (2023)
16. Muñoz González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E.C., Roli, F.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: ACM Workshop on Artificial Intelligence and Security (2017)
17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples (2014)
18. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: BadNets: evaluating backdooring attacks on deep neural networks. IEEE Access **7**, 47230–47244 (2019)
19. Guiga, L., Roscoe, A.W.: Neural network security: hiding CNN parameters with guided grad-Cam. In: International Conference on Information Systems Security and Privacy (ICISSP) (2020)
20. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)
21. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep Anomaly Detection With Outlier Exposure (2018)
22. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Netw. **4**(2), 251–257 (1991)

23. Hwang, U., Park, J., Jang, H., Yoon, S., Cho, N.I.: PuVAE: a variational autoencoder to purify adversarial examples. IEEE Access **7**, 126582–126593 (2019)
24. Jia, H., Choquette-Choo, C.A., Chandrasekaran, V., Papernot, N.: Entangled watermarks as a defense against model extraction. In: USENIX Security Symposium (2021)
25. Jin, C., Sun, M., Rinard, M.: Incompatibility clustering as a defense against backdoor poisoning attacks. In: International Conference on Learning Representations (ICLR) (2023)
26. Kariyappa, S., Qureshi, M.K.: Defending against model stealing attacks with adaptive misinformation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
27. Kolouri, S., Saha, A., Pirsiavash, H., Hoffmann, H.: Universal litmus patterns: revealing backdoor attacks in CNNs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
28. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS) (2012)
29. Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Zhang, D., Priol, R.L., Courville, A.: Out-of-distribution generalization via risk extrapolation (REx). In: International Conference on Machine Learning (ICML) (2021)
30. Kwon, H.: Detecting backdoor attacks via class difference in deep neural networks. IEEE Access **8**, 191049–191056 (2020)
31. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010). http://yann.lecun.com/exdb/mnist/
32. Li, Y., Zhu, L., Jia, X., Jiang, Y., Xia, S.T., Cao, X.: Defending against model stealing via verifying embedded external features. In: AAAI Conference on Artificial Intelligence (2022)
33. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: a natural backdoor attack on deep neural networks (2020). arXiv preprint arXiv:2007.02343
34. Liu, Y., Xie, Y., Srivastava, A.: Neural trojans. In: IEEE International Conference on Computer Design (ICCD) (2017)
35. Lotfi, S., Finzi, M., Kapoor, S., Potapczynski, A., Goldblum, M., Wilson, A.G.: PAC-Bayes compression bounds so tight that they can explain generalization. In: Advances in Neural Information Processing Systems (NIPS) (2022)
36. Lu, Z., Pu, H., Wang, F., Hu, Z., Wang, L.: The expressive power of neural networks: a view from the width. In: Advances in Neural Information Processing Systems (NIPS) (2017)
37. Lukas, N., Zhang, Y., Kerschbaum, F.: Deep neural network fingerprinting by conferrable adversarial examples. In: International Conference on Learning Representations (ICLR) (2021)
38. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems (NIPS) (2017)
39. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (ICLR) (2018)
40. Maini, P., Yaghini, M., Papernot, N.: Dataset inference: ownership resolution in machine learning. In: International Conference on Learning Representations (ICLR) (2021)
41. Nagarajan, V., Kolter, J.Z.: Uniform convergence may be unable to explain generalization in deep learning. In: Advances in Neural Information Processing Systems (NIPS) (2019)
42. Oh, S.J., Augustin, M., Fritz, M., Schiele, B.: Towards reverse-engineering black-box neural networks. In: International Conference on Learning Representations (ICLR) (2018)
43. Oliynyk, D., Mayer, R., Rauber, A.: I know what you trained last summer: a survey on stealing machine learning models and defences. ACM Comput. Surv. **55**(14s), 1–41 (2023)
44. Pal, S., Gupta, Y., Kanade, A., Shevade, S.: Stateful Detection of Model Extraction Attacks (2021)
45. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical Black-Box Attacks Against Machine Learning (2016)
46. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy (EuroS&P) (2016)

47. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions (2017). arXiv preprint arXiv:1701.06548
48. Petsiuk, V., Das, A., Saenko, K.: RISE: randomized input sampling for explanation of black-box models. In: Proceedings of the British Machine Vision Conference (BMVC) (2018)
49. Pinkus, A.: Approximation theory of the MLP model in neural networks. Acta Numer. **8**, 143–195 (1999)
50. Power, A., Burda, Y., Edwards, H., Babuschkin, I., Misra, V.: Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets (2022)
51. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
52. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-Gan: protecting classifiers against adversarial attacks using generative models. In: International Conference on Learning Representations (ICLR) (2018)
53. Schwab, P., Karlen, W.: CXPlain: causal explanations for model interpretation under uncertainty. In: Advances in Neural Information Processing Systems (NIPS) (2019)
54. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-Cam: visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision (ICCV) (2017)
55. Shafahi, A., Huang, W.R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., Goldstein, T.: Poison frogs! Targeted clean-label poisoning attacks on neural networks. In: Advances in Neural Information Processing Systems (NIPS) (2018)
56. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing Properties of Neural Networks (2013)
57. Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction APIs. In: USENIX Security Symposium (2016)
58. Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. Theory Probab. Appl. **16**(2), 264–280 (1971). https://doi.org/10.1137/1116025
59. Veldanda, A.K., Liu, K., Tan, B., Krishnamurthy, P., Khorrami, F., Karri, R., Dolan-Gavitt, B., Garg, S.: NNoculation: broad spectrum and targeted treatment of backdoored DNNs (2020). arXiv preprint arXiv:2002.08313
60. Wang, D.B., Feng, L., Zhang, M.L.: Rethinking calibration of deep neural networks: do not be afraid of overconfidence. In: Advances in Neural Information Processing Systems (NIPS) (2021)
61. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-Mnist: a novel image dataset for benchmarking machine learning algorithms (2017). arXiv preprint arXiv:1708.07747
62. Xu, H., Su, Y., Zhao, Z., Zhou, Y., Lyu, M.R., King, I.: DeepObfuscation: Securing the Structure of Convolutional Neural Networks via Knowledge Distillation (2018)
63. Xu, X., Wang, Q., Li, H., Borisov, N., Gunter, C.A., Li, B.: Detecting AI trojans using meta neural analysis. In: IEEE Symposium on Security and Privacy (S&P) (2021)
64. Xu, Y., Zhang, H.: Convergence of Deep Convolutional Neural Networks (2022)
65. Xu, Y., Zhang, H.: Uniform Convergence of Deep Neural Networks With Lipschitz Continuous Activation Functions and Variable Widths (2023)
66. Ye, H., Xie, C., Cai, T., Li, R., Li, Z., Wang, L.: Towards a theoretical framework of out-of-distribution generalization. In: Advances in Neural Information Processing Systems (NIPS) (2021)
67. Yoneda, T.: Pointwise Convergence Theorem of Gradient Descent in Sparse Deep Neural Network (2023)
68. Yu, H., Ma, H., Yang, K., Zhao, Y., Jin, Y.: DeepEM: deep neural networks model recovery through EM side-channel information leakage. In: IEEE International Symposium on Hardware Oriented Security and Trust (HOST) (2020)
69. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)

Part II
Artificial Intelligence for Critical Infrastructure Protection

Opportunities and Challenges of Using Artificial Intelligence in Securing Cyber-Physical Systems

Livinus Obiora Nweke and Sule Yildirim Yayilgan

1 Introduction

Recent years have witnessed a proliferation of systems that integrate digital technologies into physical systems, such as industrial control systems (ICS), medical devices, water systems, waste management systems, and transportation systems. This integration has led to an emerging field known as cyber-physical systems (CPS), which has attracted widespread attention. While these systems build upon and extend the capabilities of earlier technologies, they incorporate substantial enhancements driven by advancements in computing, communication, and control systems. Despite the numerous advantages offered by CPS, they are vulnerable to cyberattacks, which can physically endanger individuals and cause environmental harm [57, 58, 110, 114]. As a result, securing CPS is of critical importance to the government, academia, industry, and the general public.

As the popularity of CPS increases, so does the frequency of attacks against them. For instance, a recent disruptive ransomware attack forced the closure of the largest United States (US) fuel pipeline [132], triggering a rise in the average US gasoline price [38]. This incident underscores the far-reaching impacts of attacks on CPS and the pressing need to employ disruptive technologies such as artificial intelligence (AI) to secure these systems.

L. O. Nweke (✉)
Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Noroff Accelerate, Oslo, Norway
e-mail: livinus.nweke@ntnu.no

S. Y. Yayilgan
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
e-mail: sule.yildirim@ntnu.no

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_7

AI is an emergent, disruptive technology that is transforming every aspect of our lives, with applications spanning numerous domains. The utilisation of AI in securing CPS can enhance the detection of anomalies, mitigate security threats, predict potential component or subsystem failures, create self-managing and responsive computing systems, evaluate system security, and expedite incident responses. For example, AI can be employed in CPS environments to detect anomalies in network traffic [10, 11, 94, 120] or to monitor sensor data for signs of malicious activities [13, 90, 127, 144]. AI can also leverage vast amounts of data from multiple sources to analyse and identify potential threats [54, 122, 142]. However, the use of AI in securing CPS presents its own set of challenges. Comprehending these challenges and the opportunities that AI offers in securing CPS will enhance the security and resilience of critical infrastructures and aid in defending against emerging threats.

This chapter provides an overview of AI and cybersecurity, and it explores the opportunities and challenges associated with using AI to secure CPS. We conducted a literature review to gather existing knowledge and identify gaps in the field of AI applications for securing CPS. We performed keyword searches on multiple databases including IEEE Xplore, ACM Digital Library, PubMed, and Google Scholar, using search terms like "Artificial Intelligence", "Cybersecurity", "Cyber-Physical Systems", "AI in CPS", and others. We also sought out articles citing seminal work in this field to ensure a comprehensive exploration. The literature was analysed, themes were identified, and the opportunities and challenges of using AI for securing CPS were extrapolated.

To offer a more practical perspective, we have included case studies demonstrating successful implementations of AI in securing CPS. These case studies were selected for their exemplary integration of AI and CPS. We conducted an in-depth analysis of these selected cases, focusing on their objectives, the AI strategies used, the outcomes, and their impact on the overall security of the CPS. These case studies were sourced from publicly available documents, including technical reports and primary literature. We synthesised the lessons learnt from these cases and made relevant connections with our literature review findings.

The remainder of this chapter is structured as follows. Section 2 defines AI and discusses the broader implications of AI development beyond technical systems and challenges. It also outlines what cybersecurity entails and why cybersecurity challenges arise, examining some of the most common cybersecurity threats and their potential impacts. Section 3 discusses the opportunities that arise when using AI to secure CPS. Section 4 addresses the challenges of using AI in securing CPS. Section 5 presents case studies demonstrating successful implementation of AI in securing CPS. Section 6 summarises the key takeaways from this chapter.

2 AI and Cybersecurity

AI is a field within computer science that aims to build systems that mimic the intelligence, cognitive functions, and actions of biological beings, with the goal of

thinking and acting in a manner like humans [156]. AI is rapidly progressing and is aiding in the delivery of superior services, enhanced productivity, and high-quality outcomes to both individuals and organisations. Below is a list of sectors where AI is making significant contributions [65]:

- Marketing and advertising
- Logistics
- Energy
- Supply chain
- Cybersecurity
- Agriculture
- Real estate
- Transportation
- Education
- Entertainment
- Data analytics
- Construction
- Healthcare
- Banking and finance
- Consulting and outsourcing

As can be seen from the list above, cybersecurity is one of many areas where AI presents enormous potential. However, we are still in the early stages of harnessing this potential. The full extent of AI's capabilities in cybersecurity, and the necessity for its use, will become increasingly apparent in the coming years. Consequently, this section not only discusses the broader implications of AI development beyond technical challenges but also delineates the concept of cybersecurity, and the reasons behind the emergence of its challenges. This provides the necessary background for our exploration of the opportunities and challenges associated with using AI to secure CPS.

2.1 AI: Beyond Technical Systems and Challenges

AI has emerged as a transformative force with the potential to span various sectors and industries. While AI's ability to solve complex technical problems is undeniably impressive, addressing the broader implications of AI development beyond just technical systems and challenges is of crucial importance. This subsection will discuss four critical aspects of these broader implications: ethical implications, socioeconomic impact, legal and regulatory frameworks, and education and public engagement.

2.1.1 Ethical Implications

The design and development of AI systems must consider several ethical considerations to ensure alignment with human values and the upholding of fundamental rights [51]. These considerations arise due to the increasing integration of AI systems into our lives and their potential to significantly influence every aspect of our existence in ways we may not yet fully comprehend [105]. Thus, a comprehensive understanding of the ethical implications of designing and developing AI systems is essential to ensure that AI remains human-centric. These critical ethical considerations include fairness, transparency, privacy, and accountability.

Fairness, as an ethical consideration in designing and developing AI systems, refers to ensuring that algorithms and machine learning models do not exhibit any form of bias or discrimination against individuals or groups based on race, gender, age, or ethnicity [91]. This consideration is essential because AI systems can inadvertently perpetuate and exacerbate existing biases due to biased training data, flawed algorithms, or unjust decision-making processes [123]. To ensure fairness in AI systems, we need to implement measures that can detect and mitigate bias in training data [101]. Such measures may include regular audits and evaluations, ensuring that training data is representative of a diverse population, and developing AI algorithms that explicitly account for fairness. Including diverse stakeholders in AI development is also crucial.

Transparency in AI refers to the ability to understand and interpret the processes, decisions, and outcomes of AI models [46]. It involves making the inner workings of AI models more transparent and accessible, which will facilitate the evaluation of the reliability and accuracy of the results. Transparency is essential for building trust and ensuring that AI systems are understandable and explainable [137]. To achieve transparency in the design and development of AI systems, we need to provide comprehensive documentation of AI systems and insights into the AI decision-making process [39]. Encouraging openness in AI development, such as sharing code, data, and research, and supporting clear communication of AI system capabilities, limitations, and potential risks are also beneficial.

Another ethical consideration in designing and developing AI systems is the potential impact on individual privacy [161]. AI systems typically rely on a large amount of data, which raises concerns about individual privacy and data protection [67]. To address privacy implications in the design and development of AI systems, techniques that use the minimum amount of data necessary for AI tasks can be employed to help mitigate privacy risks [49]. Also, ensuring that data is securely stored and transmitted with appropriate access controls and encryption, employing data anonymisation techniques, and incorporating privacy considerations throughout the AI development process can help to address privacy concerns adequately.

Accountability in AI pertains to ensuring that designers and developers of AI systems comply with relevant standards and regulations [19]. As AI systems increasingly perform tasks autonomously, considering accountability mechanisms for AI systems' actions and decisions is crucial to ensure responsibility and

provide remedies in cases of misuse [70]. The critical aspects of AI accountability include developing legal frameworks that clearly define the responsibilities of AI developers, operators, and users; implementing robust monitoring and reporting mechanisms; and conducting algorithmic impact assessments.

2.1.2 Socioeconomic Impact

The socioeconomic impact of AI refers to the effect that AI has on society and the economy [103]. As a rapidly evolving technology, AI is already changing how we live and work, and its impact on society and the economy is expected to be profound. AI can transform the socioeconomic landscape, with potential consequences on the labour market, the digital divide, and opportunities for social good [75]. Therefore, as AI systems become more pervasive, understanding these impacts is essential to navigate potential challenges and harness AI's potential for positive outcomes.

One aspect of AI's socioeconomic impact is its potential effect on the labour market. AI systems can automate tasks, potentially leading to job displacement, unemployment in certain industries, and widening income inequality [69]. However, AI may also lead to new job roles and the augmentation of existing ones [50]. Assessing the types and levels of jobs at risk of automation is vital to identifying vulnerable sectors and workforce segments, which will enable targeted interventions and policies [145]. Further, developing strategies for workforce adaptation, such as retraining, upskilling, and reskilling, can help individuals transition to new job roles in an AI-driven labour market.

Another socioeconomic impact of AI is the digital divide, which is the growing gap between individuals and communities with access to AI technologies and those without [28]. This unequal access can exacerbate existing socioeconomic inequalities [89]. To address the digital divide and ensure equitable AI access, efforts to provide all populations with access to adequate digital infrastructure and promote digital literacy through education and training programs are needed [75]. Moreover, developing AI systems that cater to diverse populations, languages, and cultures and encouraging collaboration between governments, businesses, and non-governmental organisations can help make AI technologies more accessible and inclusive, ensuring the effective mobilisation of resources and expertise to address the digital divide.

AI for social good refers to using AI technologies to address global challenges and promote social well-being [48]. These efforts include initiatives to improve healthcare, address climate change, reduce poverty, improve educational outcomes, promote sustainability, and more. AI for social good programs often involves partnerships between academics, non-profits, governments, and private sector organisations to jointly address these challenges using advanced technologies [150]. Discussing methods to promote the development of AI applications for social good can help guide AI's potential towards positive outcomes.

2.1.3 Legal and Regulatory Frameworks

The design and development of AI systems must also consider legal and regulatory implications to ensure they operate responsibly and safely while promoting innovation and growth [118]. The need for appropriate legal and regulatory frameworks becomes more pressing as AI systems become increasingly integrated into various aspects of society. These frameworks would help establish rules that govern the development, deployment, and use of AI technologies to ensure that AI systems are safe and ethical, and do not infringe on individual rights. The critical aspects of legal and regulatory frameworks for AI are as follows: compliance with existing laws, AI-specific regulations, international cooperation, and industry self-regulation.

AI systems are expected to operate within the boundaries of existing legal frameworks [9]. For example, ensuring that AI systems comply with the General Data Protection Regulation (GDPR) in the European Union is essential to safeguard personal data and privacy [15]. AI-generated content and inventions also raise complex intellectual property questions, necessitating clarification of the application of existing intellectual property laws to AI systems to protect innovation and investment [25]. Additionally, AI systems must adhere to consumer protection regulations to ensure they are safe, reliable, and transparent [81]. Regulatory bodies should, therefore, be able to monitor AI systems' compliance with consumer protection laws and address potential harms.

New legal and regulatory frameworks tailored specifically to AI systems are also needed to address their unique challenges [53]. For instance, establishing guidelines and certification processes for AI systems can help build trust and promote responsible development [20]. Developing legal frameworks that assign accountability for AI systems' actions and decisions is also essential to ensure responsibility and remedy in cases of misuse [102]. Furthermore, regulations that require AI systems to provide explanations of their decision-making processes are necessary to ensure AI outputs are understandable and trustworthy [60]. Legal frameworks addressing potential biases and discrimination in AI systems are also needed to promote fairness and equitable treatment for all individuals [154].

Given that the design and development of AI systems transcend national borders, it is vital to consider international cooperation and harmonisation of AI regulations [6]. As AI systems rely on vast datasets, which may involve cross-border data flows, developing agreements and frameworks to facilitate cross-border data flows is crucial [30]. Encouraging the alignment of AI regulations across countries will help create a more predictable and consistent legal environment, fostering global collaboration and reducing barriers to designing and developing AI systems [24]. Engaging in international dialogue and collaboration on AI policy will support the sharing of best practices [115], addressing common challenges and developing joint strategies to harness AI's potential for the public good.

Industry self-regulation is another crucial aspect of legal and regulatory frameworks for AI and can play a complementary role in promoting responsible design and development of AI systems [32]. For example, developing and adopting voluntary industry guidelines can help set best practices and drive responsible AI

development [77]. Professional organisations and industry associations can also develop codes of conduct for AI developers and practitioners to promote ethical behaviour and responsible AI development. Employing industry-led certification can demonstrate compliance with ethical AI standards, fostering trust and transparency in AI systems' design and development [98]. Encouraging collaboration between stakeholders, including governments, industry, academia, and civil society, will help create more inclusive AI systems.

2.1.4 Education and Public Engagement

A crucial broader implication of AI development beyond technical systems and challenges pertains to education and public engagement in AI. This concept refers to efforts to increase public understanding and awareness of AI technologies, their applications, and their potential societal impact [159]. This engagement comprises AI literacy, interdisciplinary collaboration, public awareness, and stakeholder participation. The overarching goal is to ensure inclusive AI development and adoption that aligns with societal needs.

AI literacy refers to the fundamental competencies or knowledge about AI that the general population should possess [80]. It is vital as it enables individuals to make informed decisions regarding AI technologies and to engage in meaningful discussions about AI's societal impact [86]. To foster AI literacy, we should consider integrating AI literacy and related subjects into primary, secondary, and tertiary education curricula [106]. This integration will help build foundational knowledge and skills from an early age. Likewise, encouraging lifelong learning opportunities will help individuals continuously adapt to the evolving AI landscape [7]. AI literacy can also be bolstered by equipping educators with the knowledge and resources needed to teach AI-related subjects [106] and by fostering the development of both technical and non-technical skills, such as critical thinking, creativity, and problem-solving, to help individuals navigate the changes brought by AI in the workforce and society.

Interdisciplinary collaboration is another pivotal aspect of education and public engagement in AI. This collaboration involves experts and stakeholders from diverse disciplines, such as computer science, ethics, and social sciences, to develop AI systems that are aligned with human values and needs [92]. Interdisciplinary collaboration brings together a range of perspectives, skills, and expertise, all of which are essential for identifying and addressing the social, ethical, and policy challenges associated with AI [37]. Cultivating such collaboration will facilitate synergies between AI and other fields, leading to more robust and ethically grounded AI systems. Moreover, promoting diversity in AI research and development teams, encompassing gender, cultural, and disciplinary diversity, will help create AI systems that better cater to diverse populations and needs.

Public awareness is fundamental to education and public engagement in AI as it enhances people's knowledge and understanding of AI technologies [159]. It can help to build trust, improve understanding, and enable informed decision-making

regarding AI policy and regulation [20]. An effective approach to fostering public awareness includes collaborating with media outlets to disseminate accurate and accessible information about AI technologies, educating the public, and countering misinformation [151]. Similarly, organising public forums, conferences, and events to discuss AI developments and applications will foster dialogue and enhance public understanding of AI technologies. Moreover, creating accessible educational resources, such as articles, videos, and interactive tools, will enable the general public to learn about AI technologies and their potential societal impact.

Finally, the involvement of a wide range of stakeholders in AI development and policymaking is critical to ensure that AI systems are trustworthy, aligned with social values, and serve the public interest [159]. Stakeholders' participation can take several forms, including public consultations, workshops, conferences, online forums, and surveys. For instance, engaging the public in AI policymaking will ensure that AI policies reflect societal needs and values [95]. Similarly, encouraging collaboration between governments, industry, academia, and civil societies in AI development and policymaking will create more inclusive and effective AI strategies [152]. Furthermore, ensuring that underrepresented communities have a voice in AI development and policymaking will help address potential biases and promote the equitable growth of AI technologies [64].

2.2 What Is Cybersecurity and Why Do Cybersecurity Challenges Arise?

Cybersecurity involves safeguarding computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks, theft, damage, and unauthorised access. As a subfield of information security, it employs an array of technologies, processes, and practices aimed at ensuring the confidentiality, integrity, and availability of data in cyberspace [108]. The increasing reliance on technology and the Internet has concurrently elevated the significance of cybersecurity. This subsection explores the underlying factors contributing to the escalation of cybersecurity challenges.

The escalation in cybersecurity challenges in recent years can be attributed to various factors. One of these factors is increased connectivity, defined by the escalating interconnectedness of devices, systems, and networks that enables seamless communication and data sharing [73]. The drive for this development has been technological advancements and the burgeoning global Internet adoption. However, this expanded connectivity exposes vulnerabilities and offers more opportunities for cybercriminals to penetrate networks and compromise data.

Another contributing factor to the surge in cybersecurity challenges is the growing number of Internet users [162]. As the Internet continues to expand its reach, more individuals worldwide are accessing the web. This increased usage has led to a commensurate rise in sensitive online information, ripe for exploitation

by cybercriminals for financial gain, identity theft, or other malevolent activities [8]. Moreover, many new Internet users may lack the technical knowledge to adequately protect themselves from online threats, thereby becoming easy targets for cyberattacks.

Additionally, cyberattacks have grown more sophisticated in recent years. Cybercriminals continuously refine their tactics and devise new methods to evade security controls. The deployment of advanced techniques such as AI has led to the inclusion of a new type of threat (AI abuse) in the European Union Agency for Cybersecurity's emerging cybersecurity threats and challenges for 2030 report [44]. This further complicates the task for cybersecurity professionals in detecting and responding to cyber threats. Social engineering tactics, now commonplace [104], are leveraged to exploit human vulnerabilities and gain unauthorised access to sensitive information.

Furthermore, the rise in cybersecurity challenges can be attributed to our increasing dependence on mobile networks and devices [17]. The widespread use of smartphones and tablets has led to storing personal and professional information on these devices [62], making them attractive targets for cybercriminals. Public Wi-Fi networks, application-based vulnerabilities, and mobile malware have all contributed to an increased risk of cyberattacks on mobile devices.

In summary, cybersecurity is a critical facet of our modern life, tasked with protecting computing devices, networks, and most importantly, data and information from cyber threats. The challenges associated with cybersecurity arise from a combination of factors, including increased connectivity, a growing number of Internet users, more sophisticated attacks, and a greater reliance on mobile networks and devices. Addressing these challenges requires a comprehensive approach, incorporating appropriate technologies, processes, and practices and fostering a culture of security education and awareness.

2.3 Cybersecurity Threats and Impact of Threats

Cybersecurity threats are events or activities that jeopardise the confidentiality, integrity, or availability of information systems. These threats can result in unauthorised disclosure, misuse, alteration, or destruction of information or information systems [111]. Attackers continuously innovate, developing new techniques to exploit information systems and process vulnerabilities. This subsection explores common cybersecurity threats and their potential impacts.

- Malware is software designed to infiltrate, damage, or disrupt a computer system or network. Common types include viruses, Trojans, worms, spyware, and ransomware. Attackers often use social engineering techniques, such as phishing emails, to coax users into downloading malware.
- Phishing is a social engineering attack using fraudulent emails, text messages, or websites to trick users into revealing sensitive information or downloading

malware. These communications often impersonate legitimate entities, making their malicious intent difficult to discern.
- Denial of service (DoS) and distributed DoS (DDoS) attacks overwhelm a network or website with traffic, rendering it inaccessible. Such attacks can cause significant disruptions, resulting in lost revenue and reputational damage.
- Advanced persistent threats (APTs) are targeted attacks that utilise advanced techniques to breach specific organisation defences and stay undetected for an extended period.
- Insider threats originate from individuals with legitimate access to an organisation's systems, including employees, contractors, or partners. These threats can be either intentional or unintentional.
- SQL injection attacks exploit vulnerabilities in web applications to access or modify sensitive information. Such attacks often result from a lack of proper input validation.
- Man-in-the-middle (MitM) attacks intercept communications between two parties to eavesdrop or alter the content, leading to theft of sensitive information.
- Zero-day attacks exploit vulnerabilities unknown to the affected software's vendor or developer. The absence of available patches or fixes during the attack leaves systems vulnerable.
- Cryptojacking involves unauthorised use of a victim's device for cryptocurrency mining, leading to financial losses due to increased energy consumption and device wear and tear.
- Botnets are networks of compromised computers controlled by an attacker, often used for DDoS attacks or other malicious activities.

The impact of cybersecurity threats is broad and severe, affecting individuals, organisations, governments, and society at large. Individuals may experience financial loss, identity theft, loss of privacy, and emotional distress [3]. Organisations face threats such as financial loss, operational disruption, intellectual property theft, regulatory non-compliance (potentially resulting in fines, sanctions, and increased regulatory scrutiny), and reputational damage [34].

Furthermore, cybersecurity threats can have national security implications [121], especially when nation-state actors are involved. High-profile breaches can undermine trust in government institutions, eroding public confidence in the government's ability to protect sensitive information and maintain national security. Broader societal consequences can also arise from attacks on critical infrastructure, including power grids, hospitals, water systems, waste management systems, and transportation systems [113]. These attacks can cause widespread disruption and even loss of life.

In conclusion, the impact of cybersecurity threats extends beyond the immediate financial implications of a breach. The ripple effects include reputational damage, operational disruption, legal and regulatory consequences, and national security concerns. Hence, harnessing the advantages of AI in combating cyber threats and mitigating their impacts, especially in securing CPS, becomes essential.

3 Opportunities of Using AI in Securing CPS

This section explores the opportunities arising from the use of AI in securing CPS. We will delve into various areas such as anomaly detection, predictive maintenance, intrusion detection and prevention, autonomic and adaptive systems, security assessment and penetration testing, and incident response and recovery, all of which demonstrate how AI can enhance the security, resilience, and performance of CPS. By leveraging these opportunities, AI can significantly contribute to CPS's security and robustness, helping to protect critical infrastructure and reduce the risk of catastrophic failures. However, we must also acknowledge that the incorporation of AI introduces new challenges, which we will analyse in the subsequent section.

3.1 Anomaly Detection

AI holds considerable potential to enhance anomaly detection and mitigate security threats within CPS. Specifically, the use of AI for anomaly detection is becoming an increasingly crucial tool for ensuring the security and reliability of CPS [129]. This trend is driven by the increasing complexity and unique attributes of CPS, which often render traditional security methods insufficient for detecting anomalies and mitigating security threats [109]. Therefore, employing AI for anomaly detection offers an attractive alternative capable of identifying unusual patterns, behaviours, or events, ensuring the ongoing performance and resilience of CPS.

Several recent methods use AI for anomaly detection in CPS. For instance, the authors in [76] describe an anomaly-based approach for detecting and classifying attacks in CPS, which uses anomaly detection to define a model for normal system behaviour and a supervised attack model to classify anomalies. The findings from this study show that the Naïve Bayes classifier used to train the attack model could detect and classify attacks with satisfactory accuracy. A comparison of the performance of several machine learning models, such as logistic regression, support vector machine, decision tree, random forest, and artificial neural network, used to predict attacks and anomalies in CPS is provided in [63]. The authors suggest that while most of these approaches have similar accuracy, other metrics indicate that the random forest model performs comparatively better.

The authors in [88] review deep learning-based anomaly detection methods in CPS, discussing their limitations and deficiencies to improve their design and evaluation. They identify several opportunities for using deep learning-based anomaly detection methods in CPS, including handling CPS's increasing complexity, learning from large volumes of data without requiring domain-specific knowledge, adapting to the changing CPS environment to detect new attacks, providing high detection accuracy and low false positive rates, and potentially integrating with other security mechanisms in the CPS environment to provide comprehensive protection.

Recent studies have also highlighted the promising potential of using AI for anomaly detection in CPS. For example, the study in [71] presents an intelligent anomaly detection in CPS, which uses data-driven AI tools employing multi-class support vector machines. This work considers the effects of cyber anomalies, such as false data injection, DoS attacks, and physical anomalies due to power system faults. The authors in [61] propose a solution for real-time ICS network traffic anomaly detection. Their proposed hybrid statistical-machine learning model integrates a seasonal autoregressive integration moving average (SARIMA)-based dynamic threshold model and a long short-term memory (LSTM) model to jointly identify abnormal traffic patterns with low false omission rates. The study in [116] proposes a novel detection framework for CPS based on error space reconstruction, using genetic algorithms to perform hyper-parameter optimisation of machine learning methods. The results from this work show that the proposed framework can effectively detect anomalies in CPS, providing a more secure infrastructure.

The opportunities provided by using AI for anomaly detection in CPS could significantly enhance the security, reliability, and efficiency of these systems. Much research is ongoing to develop and refine the existing methods for practical applications. Thus, leveraging AI to detect unusual patterns, behaviours, or events in the CPS environment can proactively address potential security threats and operational issues, ensuring the security and stability of CPS.

3.2 Predictive Maintenance

AI holds significant potential for predictive maintenance in CPS. Predictive maintenance is a proactive maintenance approach designed to predict when components or subsystems might fail, thereby allowing timely maintenance to minimise downtime, enhance efficiency, and reduce maintenance costs [1]. This approach leverages data from sensors, historical records, and other sources to identify trends, patterns, and correlations that may signify potential failure. Given the severe consequences of component or subsystem failure in CPS, AI methods can be instrumental in building models that identify potential problems before they occur.

Machine learning techniques have been utilised for predictive maintenance in CPS. For instance, the authors in [31] propose a hybrid machine learning approach, merging supervised and semi-supervised learning for predictive maintenance. This approach can integrate heterogeneous machine data to offer insights for the effective management of CPS. The authors in [119] describe a machine learning architecture for predictive maintenance based on the random forest approach. They tested the proposed architecture on a real industry example, and preliminary results suggest that the approach effectively predicts different machine states with high accuracy.

In recent years, a variety of AI approaches have been employed for predictive maintenance in CPS. For example, the authors in [27] introduce a data-driven approach to predictive maintenance, where they calculate predicted failure probabilities using tree-based classification models and the temporal evolution of event

data. This approach provides insightful information for maintenance management practices. A convolutional neural network (CNN) algorithm has been used for predictive maintenance, with its performance evaluated in fault analysis of a predictive maintenance system [100]. The results suggest that it could be a powerful tool for multi-class fault detection.

Furthermore, the authors in [18] propose a deep learning model for predictive maintenance in CPS using long short-term memory (LSTM) autoencoders. They use this AI approach to classify real-world machine and sensor data to estimate the remaining useful life of monitored systems. Their work suggests that redundant and preventive stoppage in the production line can be minimised, concurrently reducing maintenance operation costs. Similarly, the study in [141] proposes a five-layer CPS framework for smart production lines, integrating physics and data-driven methods. The authors believe this framework can create a closed-loop workflow, reducing potential failures that could impact smart production line operations. It can also be generalised to predict remaining useful life and product quality.

The use of AI for predictive maintenance has the potential to substantially enhance the efficiency, reliability, and performance of CPS. Its role is not confined to preserving the physical integrity of the system; it also significantly contributes to cybersecurity. This is because unplanned system downtime can create windows of opportunity for cyberattacks. Therefore, by leveraging data from sensors, historical records, and other sources, it is anticipated that AI will become increasingly prevalent in CPS for anticipating and addressing potential issues before they escalate.

3.3 Intrusion Detection and Prevention

The application of AI in intrusion detection and prevention within CPS has proven to be promising. AI-driven intrusion detection and prevention systems, designed to identify and respond to suspicious behaviour or attacks in real time, can reduce the potential impact of cyberattacks on systems [94]. These systems typically utilise machine learning algorithms to recognise and categorise patterns indicative of attacks. Deep learning and neural network techniques can be employed to further enhance the accuracy and efficiency of intrusion detection and prevention systems in CPS.

AI-driven intrusion and prevention systems provide a promising avenue to bolster CPS security, reducing the risk of costly attacks. For instance, an AI-based network intrusion detection system with hyper-parameter tuning has been proposed, with an impressive performance accuracy score of 99.97% and an average area under the ROC curve of 0.999 [68]. Similarly, the study in [93] describes the successful application of machine learning models to detect attacks in CPS with high accuracy, efficiency, and detection rates. The author argues that these models outperform traditional methods, achieving an accuracy of over 90%.

The authors in [120] propose a neural network-based architecture capable of detecting and isolating integrity cyberattacks in CPS. Their simulation results show high effectiveness in identifying and locating denial of service (DoS) and integrity attacks. The study in [11] presents an intelligent cognitive computing-based intrusion detection system for CPS, featuring a novel binary bacterial foraging optimisation (BBFO)-based feature selection technique and a gated recurrent unit (GRU) model classifier for intrusion detection. The results demonstrate promising performance, with an accuracy of 98.45%.

Moreover, the authors in [10] introduce a novel AI-enabled fusion-based intrusion detection system for CPS. Their system employs data pre-processing techniques including data conversion and normalisation, enhanced with a fish swarm optimisation-based feature selection technique, and a multi-model fusion of three models using a weighted voting-based ensemble technique. Simulation analysis shows an improvement over existing method. The study in [94] presents an AI-based optimisation with a deep learning model for blockchain-enabled intrusion detection in a CPS environment. The author contends that this AI-based intrusion detection approach holds great promise in improving security in CPS environments.

The application of AI for intrusion detection and prevention holds significant potential to enhance CPS security. It can identify and respond to potential threats more accurately, adaptably, and efficiently. Further research and development in this field are necessary to fully exploit the opportunities presented by AI-enabled intrusion detection and prevention in CPS, thereby ensuring the resilience of CPS against evolving threats.

3.4 Autonomous and Adaptive Systems

AI holds immense potential in transforming autonomous and adaptive systems in CPS. Such systems are characterised by their ability to self-manage and dynamically adjust to changes in their environment or requirements without requiring human intervention [35]. Given the increasing complexity and scale of modern CPS, designing these systems to be resilient, efficient, and responsive is crucial. The use of AI in autonomous and adaptive systems can enable CPS to manage themselves and adapt to new threats or environmental changes autonomously.

Various ongoing research initiatives aim to explore the potential of AI-enabled autonomous and adaptive systems in CPS. For instance, the study in [66] proposes a self-adaptation approach for autonomous CPS, employing machine learning to identify Pareto-optimal configurations. The authors argue that this approach facilitates automated runtime decision-making for self-adaptation of highly configurable systems, allowing a deeper exploration of solution spaces that would otherwise be intractable. Independent evaluations show that this approach results in high-quality adaptation plans in uncertain and adversarial environments.

The authors in [40] present a learned Monitor, Analyse, Plan, Execute, and Knowledge (MAPE-K)-based model to support self-adaptation in CPS. They

employ fully supervised machine learning to train the traditional MAPE-K model to recognise normal system behaviour, generating alerts to ensure CPS's reliability, flexibility, and protection against cyber threats. Similarly, the study in [4] proposes a real-time adaptive sensor attack detection framework comprising three components: an attack detector, a behaviour predictor, and a drift adaptor. The authors demonstrate the framework's efficiency and efficacy using realistic sensor data from autonomous CPS, showing its ability to adjust its behaviour to meet attack deadlines.

As observed, the use of AI for autonomous and adaptive systems in CPS presents numerous opportunities. It can facilitate the creation of self-managing and responsive computing systems within CPS environments, thereby improving the performance, resilience, and efficiency of CPS. These capabilities are particularly valuable given the rapid evolution of technologies, escalating system complexity, and the rising significance of cybersecurity.

3.5 Security Assessment and Penetration Testing

Security assessment and penetration testing, leveraging AI, are burgeoning fields with a multitude of potential applications, including in CPS [99]. These proactive strategies for system security evaluation help to unearth vulnerabilities and potential attack vectors. In the context of CPS, AI significantly enhances the effectiveness of security assessment and penetration testing by automating the process, which subsequently reduces the cost and time associated with identifying and rectifying vulnerabilities. For instance, the authors in [97] utilise AI for cybersecurity risk assessment in CPS, clarifying cyber risks and impacts in real-world scenarios while suggesting corresponding countermeasures.

The power sector, a key application area within CPS, can benefit immensely from AI for security assessment and penetration testing. For instance, the authors in [84] propose a deep reinforcement learning-based penetration testing framework, designed to efficiently pinpoint critical vulnerabilities in smart grids. Simulation results demonstrate that this approach effectively identifies the optimal attack path against system stability under conditions of high load demand, solar power generation, and weather variations. This represents an encouraging progression towards a highly customisable framework for complex CPS penetration testing, involving automatic deep reinforcement learning agents and diverse attack schemes. Similarly, the authors in [135] employ deep learning models for security assessment in smart grids, showing efficacy in detecting false data cyberattacks.

The study in [21] introduces a new domain-aware reinforcement learning approach for automated adversary emulation within CPS. The authors' performance demonstrations highlight the solution quality of proposed algorithms on a use case involving sensor deception attacks on buildings, indicating that automated adversary emulation provides a comprehensive assessment of CPS resilience against cyberattacks. Furthermore, the authors in [130] propose a security assessment

and penetration testing framework that utilises machine learning techniques on an ensemble of regular expressions to generate new attack vectors and security vulnerabilities. The framework also incorporates a defence-in-depth mechanism for detecting known attacks and possible novel exploits, optimising the cost of security measures based on the sensitivity of protected resources.

As crucial components of a comprehensive cybersecurity strategy, both security assessment and penetration testing can benefit immensely from AI enablement. By automating vulnerability scanning, enhancing threat intelligence, and improving the accuracy and efficiency of various testing methodologies, AI-enabled security assessment and penetration testing can significantly reduce the likelihood of successful cyberattacks and minimise the potential impact of security incidents.

3.6 Incident Response and Recovery

AI has profound potential in enhancing incident response and recovery strategies in CPS, which are crucial components of a comprehensive cybersecurity strategy, primarily focusing on managing and mitigating the effects of security incidents [143]. AI methods can be employed to devise effective incident response and recovery plans in CPS, aiming to minimise damage, shorten recovery time, and reduce associated costs. A notable example is the work in [41], which employed a Bayesian inverse reinforcement learning technique to predict sensor spoofing attack goals, identify compromised sensors, and restore the system.

AI also plays a pivotal role in autonomously mitigating cybersecurity risks within the CPS environment. For instance, the authors in [72] utilise a hierarchical risk correlation tree to model an attacker's pathways towards specific goals and employed a competitive Markov decision process to model the reciprocal security interaction between the protection system and the attacker. The experimental results demonstrated that the autonomous response controller could effectively respond to single line to ground attacks, thereby recovering the CPS. Similarly, the authors in [5] propose a data-driven attack recovery framework that restores CPS from sensor attacks, leveraging natural redundancy among heterogeneous sensors and historical data for attack recovery. The results showed the framework's effectiveness in maintaining system functionality in the face of sensor attacks.

Moreover, AI can significantly support decision-making and enhance threat intelligence during incident response and recovery in CPS. An illustration of this is provided in [82], which outlined the use of AI for incident response in CPS, using an architecture of intelligent IoT to coordinate with ground, surface, aerial, and underwater robots for large-scale environmental data collection and decision-making for response. Similarly, the study in [155] proposed an AI-enabled CPS model to identify potential threats and enhance incident response and recovery. The authors reported that their proposed model could improve accuracy, prediction, packet loss, and latency, providing cybersecurity authorities with a potent tool to combat the continuously evolving attacks posed by adversaries.

In conclusion, AI's application in incident response and recovery can significantly enhance the detection, analysis, response, and recovery from security incidents. Given the increasing importance of robustness and security of CPS in critical infrastructures, such enhancements are vital.

4 Challenges of Using AI in Securing CPS

This section presents an analysis of five critical challenges related to the use of AI in securing CPS: data quality and availability, vulnerability to adversarial attacks, lack of transparency and interpretability in AI models, limited understanding of AI by security practitioners, and ethical and legal considerations. We will highlight various contributions aimed at addressing these challenges and provide a critical review of open issues that warrant further investigation.

4.1 Data Quality and Availability Issues

Data quality and availability pose significant challenges to the effective use of AI in securing CPS [12, 122]. "Data quality" refers to the accuracy, completeness, reliability, and relevancy of the data used to train AI systems. Factors such as incomplete, inaccurate, or biased data can adversely impact it [131, 157]. Incomplete data may lead to gaps in analysis and decision-making, inaccurate data can result in incorrect conclusions, and biased data may perpetuate existing biases.

"Data availability" concerns the presence of sufficient and relevant data for training AI systems. It is crucial for developing precise and efficient AI models suitable for CPS [146]. However, data availability can be a challenge for CPS as certain systems may not produce enough data or may not generate data in a format suitable for analysis [12]. Additionally, privacy and data protection requirements may restrict the availability of data needed to secure CPS [125].

To address these concerns, strategies such as data cleaning and normalisation techniques can be implemented to improve data quality. Concurrently, data governance policies and procedures can be put into place to ensure data accuracy, completeness, and unbiasedness. Data availability issues can be mitigated by strategies such as data-sharing agreements and investing in technologies like sensors and Internet of things (IoT) devices to increase the volume of data available for analysis. Notably, significant efforts have been made to address these issues [59, 79, 136], but several questions remain:

- How can we ensure the accuracy and representativeness of data used to train AI models for securing CPS?
- How do we address data sharing and dissemination concerns in the context of AI-based CPS security?

- How can the privacy and security of collected data be guaranteed in the context of using AI in securing CPS?
- What legal and regulatory frameworks need to be established to facilitate data sharing and dissemination in this context?

Recent work has begun to answer these questions. For instance, a dSPACE hardware-in-the-loop (HIL) simulator was used to collect the HIL-based augmented ICS security (HAL) dataset 1.0, the first CPS dataset collected using the HAL testbed [139]. Similarly, the authors in [55] propose a methodology to generate reliable anomaly detection datasets in CPS. In [138], the implementation of a programmable CPS testbed for anomaly detection is discussed, with plans to develop and release CPS datasets using the testbed in the future.

The legal analysis in [112] reviews how laws and regulations support or refute cyber threat intelligence sharing, providing guidance for entities participating in information sharing for critical infrastructure protection. Additionally, the study in [14] discusses the legal protection of AI data and algorithms from the perspective of IoT resource sharing, noting a lack of privacy protection laws in the current AI data-sharing environment.

While these initial answers partially address the aforementioned questions, further efforts are required. Given that the performance of AI systems heavily depends on the data they are trained on, it is essential to address data quality and availability issues to effectively use AI in securing CPS.

4.2 Vulnerability of AI Models to Adversarial Attacks

The susceptibility of AI models to adversarial attacks is a notable challenge in the application of AI for securing CPS. An adversarial attack is a calculated attempt to manipulate or deceive an AI model by introducing maliciously designed data during its training or testing phase. The objective is to make the AI system yield incorrect predictions or decisions, which could potentially lead to severe consequences.

Various forms of adversarial attacks can target AI models. For instance, in a poisoning attack, an attacker intentionally modifies or inserts training data to induce errors or biases into the model [149]. In an evasion attack, the attacker alters the input data being processed by the AI system, making it challenging for the system to correctly classify or identify threats [133]. Lastly, during an inference attack, the attacker aims to deduce sensitive information from the AI model, potentially resulting in privacy violations [87].

The use of AI in securing CPS can be significantly affected by the vulnerability of AI models to these adversarial attacks. In the context of CPS, even minor prediction or decision errors can lead to serious implications. For example, an attacker might utilise an evasion attack to mislead a security system in a CPS environment into overlooking an intrusion [133], or employ a poisoning attack to trick the system into accepting a false positive as a legitimate threat [148]. In [16], the authors

demonstrate an adversarial attack on an ICS, illustrating that an adversarial attack can undermine supervised models by generating adversarial samples, thus reducing classification performance.

Numerous efforts have been directed towards addressing the vulnerability of AI models to adversarial attacks. A comprehensive review of various techniques for mitigating adversarial attacks on AI systems is presented in [117, 124]. Nonetheless, several critical questions concerning the CPS environment remain:

- What specific vulnerabilities in AI models make them prone to adversarial attacks within CPS?
- How can we enhance the security and reliability of AI models used in CPS to mitigate adversarial attack risks?
- How does the susceptibility of AI models to adversarial attacks influence the adoption of AI within the CPS environment?
- What is the potential role of regulations and standards in ensuring the security and reliability of AI models used in CPS against adversarial attacks?

4.3 Lack of Transparency and Interpretability in AI Models

Transparency and interpretability are critical factors for the effective deployment of AI models in securing CPS. Typically, AI models involve complex algorithms and are trained on extensive data, leading to challenges in understanding the logic behind their decision-making or predictive capabilities [26]. This lack of transparency and interpretability can make it hard to identify model errors or biases and can significantly undermine user trust in the system's outcomes.

In the context of CPS, clear comprehension of the decision-making process is pivotal. The lack of transparency and interpretability, therefore, presents a substantial hurdle to the successful application of AI in securing CPS [85]. Given the high-stakes nature of CPS, it is crucial that the AI models used make accurate and unbiased decisions. Thus, there is a demand for methods that can enhance the transparency and interpretability of AI models employed in securing CPS.

Various strategies have been proposed to augment the transparency and interpretability of AI models. For instance, explainable AI (XAI) refers to AI systems that elucidate their decision-making process in a format understandable by humans [56]. XAI encompasses techniques like visualisations, natural language explanations, or decision trees, all aimed at shedding light on the AI decision-making process. A comprehensive overview of existing XAI techniques, trends, and primary research directions is provided in [2].

Model-agnostic interpretation is another approach to boost transparency and interpretability. It decouples explanations from the model [128] and facilitates understanding of why AI models make specific predictions. This comprehension aids in identifying potential biases or errors and ensures more informed decisions

based on the predictions. Moreover, it can yield more intuitive user interfaces, enhancing user trust and confidence in AI systems.

Despite existing efforts to tackle transparency and interpretability issues in AI models, several crucial questions must be addressed to effectively utilise AI models in securing CPS:

- How can we ensure that AI models used in the CPS environment are trustworthy and reliable, and how do we verify their performance over time?
- What is the required level of transparency and interpretability for AI models employed in securing CPS, and how can we achieve this?
- How can we ensure that AI models do not produce unintended consequences that could jeopardise CPS security?

4.4 Limited Understanding of AI by Security Practitioners

The application of AI in cybersecurity is a relatively new development, and as a result, many security professionals may not yet possess a comprehensive understanding of AI and its potential applications [134]. This knowledge gap can contribute to a lack of confidence in AI capabilities and inhibit the recognition of AI's potential benefits in securing CPS. Furthermore, this lack of understanding could hinder the successful integration of AI into existing security frameworks and processes and impede the evaluation of AI effectiveness in securing CPS.

As AI's role in security continues to expand, the limited understanding of AI by security practitioners could pose significant obstacles to its application in securing CPS [158]. This gap could create deployment challenges for AI-based solutions within the CPS environment. Therefore, it is crucial for security practitioners to deepen their understanding of AI, enabling them to exploit its potential fully while acknowledging its associated risks and limitations.

Addressing this understanding gap requires focused efforts on education and training for security practitioners [22]. They need to be familiarised with the fundamental concepts of AI, its potential security applications, and the best practices for its integration into security processes.

Another strategy to bridge this knowledge gap involves the formation of cross-functional teams, comprising security professionals and AI experts [36]. These teams can collaborate on security projects, ensuring effective integration of AI into security processes and enhancing the understanding of AI's functionality among security professionals. As a result, these collaborations would increase the ability of security practitioners to leverage AI effectively in securing CPS and staying ahead of emerging security threats.

4.5 Ethical and Legal Considerations

The application of AI in securing CPS brings to the fore several ethical and legal considerations, warranting careful attention to ensure fairness and accountability. One significant ethical consideration is the potential impact on individual privacy [161]. AI systems usually require vast amounts of data, often including personal information, to make decisions or predictions. This requirement raises privacy and data protection concerns [15]. Consequently, when using AI in securing CPS, it is imperative to be transparent about personal data collection, storage, and usage and to adhere strictly to applicable privacy regulations.

Another key ethical concern involves the potential for bias and discrimination in AI systems [47]. As it is often stated, AI systems are only as good as the data they are trained on. Therefore, if training data is incomplete, inaccurate, or biased, the AI model can produce unfair or prejudiced decisions. This emphasises the need for ensuring unbiased data training for AI models and designing them to mitigate potential biases and discrimination.

Further, AI systems can cause certain ethical issues [33]. For instance, the deployment of AI systems in the CPS environment might lead to unemployment, as human-performed jobs get replaced [29]. This change could result in an unfair wealth distribution as our current economic system is predicated on compensating economic contributions. As AI application in the CPS environment increases, the human workforce's reliance could significantly reduce, potentially concentrating revenue among fewer individuals.

From a legal viewpoint, compliance with applicable laws and regulations is crucial for AI systems [23]. Despite the lack of well-defined laws specifically addressing potential harms caused by AI systems, AI's use in security could still be subject to specific legal requirements, such as those related to privacy and data protection [45]. Awareness of these requirements is essential for ensuring compliance when using AI to secure CPS.

Additionally, potential legal liabilities associated with using AI in securing CPS should be considered [74]. For example, should an AI system error lead to a security breach or another adverse outcome, accountability for any arising damages would need to be established [126]. It is crucial to understand the potential risks associated with AI usage in securing CPS and to implement measures to mitigate these risks, like setting up appropriate safeguards and insurance coverage.

Addressing the ethical and legal considerations associated with AI usage in securing CPS necessitates careful consideration of several questions:

- What kind of ethical and legal frameworks need to be established to support AI's use in securing CPS?
- How can we design AI to operate within the ethical and legal frameworks of CPS security?
- How can AI be used to enhance the resilience and adaptability of CPS against cyber and physical security threats, while maintaining ethical and legal considerations?

- How can we evaluate AI's use for CPS security in terms of its effectiveness, efficiency, societal impact, and adherence to ethical and legal considerations?

Efforts have been made in recent years to address some of these questions. For example, the authors in [160] discuss three types of harm that can arise from AI: individual harm, collective harm, and societal harm, arguing that societal harm, often overlooked, is not reducible to the two former types of harm. The work in [140] considers the ethics of AI and how to build ethical AI. The authors in [147] argue that it is time to work towards concrete policies for ethical and socio-legal governance in AI, within the context of existing moral, legal, and cultural values.

Several governments worldwide are also making efforts towards concrete policies for ethical and socio-legal governance in AI. For instance, in 2018, the European Commission set up an independent expert group on AI to advise on its AI strategy, including ethics guidelines for trustworthy AI [42]. The author in [78] uses a socio-legal perspective to analyse the ethics guidelines for trustworthy AI as a governance tool in AI development and use. The work in [153] assesses the ethics guidelines for trustworthy AI and notes that a law enforcement approach must be the essential next step towards the beneficial and humane development of AI.

The European Union is currently crafting a new legal framework (AI Act) [43], which aims to substantially bolster regulations on the development and use of AI. The AI Act primarily focuses on strengthening rules around data quality, transparency, human oversight, and accountability. It also aims to address ethical questions and implementation challenges in various sectors, including healthcare and energy, where CPS is mainly deployed.

5 Case Studies Demonstrating Successful Implementations of AI in Securing CPS

This section examines case studies demonstrating successful implementations of AI in securing CPS. These case studies were chosen based on their exemplary demonstration of AI and CPS integration. We conducted an in-depth analysis of the selected cases, focusing on their objectives, the AI strategies employed, the resulting outcomes, and their overall impact on CPS security. These case studies underscore the benefits and potential of AI-driven security solutions in protecting critical energy infrastructure and enhancing operational efficiency.

The study in [52] proposes a deep learning-based solution for detecting and characterising time delay attacks (TDAs) in CPS. The authors contend that TDAs pose a significant threat to CPS security, exploiting vulnerabilities in communication channels to disrupt the system's operation. They propose a hierarchical long short-term memory (H-LSTM) model that processes raw data streams from relevant CPS sensors online and continually monitors embedded signals to detect and characterise attacks. The model, trained on a simulated TDA dataset in two CPS: a power plant control system (PPCS) and an automatic generation control

(AGC) system, is compared with other machine learning algorithms like k-nearest neighbours (kNNs) and random forests (RF). The study shows that the H-LSTM model outperforms these algorithms in terms of accuracy and mean absolute error (MAE) for TDA detection and characterisation. The authors assert that their model is practical for real systems and can be configured to meet specific user requirements. This study provides a significant demonstration of successful AI application in securing CPS through a novel deep learning-based method for TDA detection and characterisation.

Vermont Electric Cooperative (VEC), a significant energy provider in Vermont, successfully collaborated with Nozomi Networks to integrate AI techniques, enhancing their power grid operations' reliability, cybersecurity, and operational efficiency [107]. VEC incorporated Nozomi Networks Guardian—an AI-driven cybersecurity solution that leverages machine learning algorithms for anomaly detection, intrusion detection and prevention, and incident response—into its existing energy infrastructure and control systems, achieving seamless cooperation and communication with legacy systems, devices, and protocols. This process involved extensive testing, validation, and training of utility personnel for effective management and interpretation of AI-generated insights. The deployment of the AI-driven cybersecurity solution at VEC yielded several significant results [107]:

- Enhanced security profile: Guardian significantly improved VEC's security profile by identifying and analysing various cyber threats, reducing successful cyberattack likelihood, and minimising potential operational impact.
- Boosted operational efficiency: Guardian decreased ICS administration and cybersecurity hours, allowing the security teams to tackle more complex challenges and reducing the costs associated with energy infrastructure security.
- Labour hour reduction: The automation of numerous security tasks by Guardian led to a weekly reduction of "10 to 12 hours" [12.5 labour weeks annually] in labour, freeing up resources for other crucial tasks.
- Reduced repair truck rolls: Guardian enabled VEC to reduce the number of repair truck rolls, thus decreasing costs and optimising resource allocation.

The study in [83] proposes a federated deep learning scheme for detecting and mitigating cyber threats against industrial CPS. The study addresses the challenge of detecting such threats, which are becoming increasingly attractive targets for state-sponsored or affiliated actors. The authors put forward a novel federated learning framework for multiple industrial CPS, enabling the collective creation of a comprehensive intrusion detection model in a privacy-preserving manner. They develop a novel CNN-GRU-based intrusion detection model capable of effectively detecting various types of cyber threats against industrial CPS. They design a Paillier-based secure communication protocol for the federated learning framework to preserve the security and privacy of model parameters during training. The study outcomes demonstrate that DeepFed achieves high accuracy in detecting cyber threats in industrial CPS while preserving the privacy and security of sensitive data, making a significant contribution to industrial CPS cybersecurity.

Furthermore, the study in [96] presents a successful AI implementation in securing CPS. The authors design a fully virtualised testing environment for data-driven analysis of cyber-physical disturbances, considering cyber and physical data simultaneously. The study's primary objective is to develop a holistic analysis approach that assesses the impact of information and communication technologies in ICS, providing a testbed environment to run cyber-physical scenarios in a controlled virtualised setting. The study employs several AI strategies, including machine learning algorithms for anomaly detection, network analysis tools, and virtualisation technologies. The results demonstrate that the proposed testbed can detect anomalous system behaviour, which can help prevent the harmful effects of cyberattacks on physical assets.

These case studies illustrate the potential of AI techniques in addressing cybersecurity challenges and improving operational efficiency in CPS. The effective use of the H-LSTM model in detecting and characterising TDAs in real-time scenarios, successful deployment of Nozomi Networks Guardian at VEC, the federated deep learning scheme for detecting and mitigating cyber threats against industrial CPS, and the fully virtualised testing environment for data-driven analysis of cyber-physical disturbances further validate the findings of the literature review presented in this chapter. By showcasing the real-world impact of AI-driven security solutions, these case studies offer valuable insights and inspiration for further research, innovation, and collaboration in the field of CPS security.

6 Conclusion

The application of AI techniques in securing CPS presents a myriad of opportunities, but also comes with certain challenges. Opportunities include enhanced threat detection and prevention, predictive maintenance capabilities, accelerated incident response and recovery, the development of self-managing and responsive computing systems, comprehensive security assessment, and optimised security-aware operations. On the other hand, challenges encompass ensuring data quality and availability, susceptibility to adversarial attacks, transparency and interpretability issues in AI models, limited understanding of AI by security practitioners, and ethical and legal considerations.

For the responsible and effective employment of AI in securing CPS, addressing these challenges and capitalising on the opportunities is paramount. This necessitates guaranteeing data quality and availability, integrating explainability and transparency into AI models, providing training and education for security practitioners on AI, and establishing ethical and legal frameworks for AI applications in CPS security. The successful utilisation of AI in securing CPS also requires a collaborative effort among security experts, AI developers, policymakers, and civil society organisations. By collectively harnessing the opportunities and tackling the challenges, stakeholders can foster a more secure and resilient future for CPS across various industries.

In conclusion, while the potential of AI to enhance the security of CPS is significant, realising this potential requires concerted efforts to confront the inherent challenges. Ongoing research, innovation, and collaboration in the field of CPS security will play a pivotal role in overcoming these challenges and pave the way for a safer, more secure world. This chapter contributes valuable insights and provides inspiration for further inquiry and exploration, underscoring AI's transformative role in securing CPS and propelling progress in the future.

References

1. Achouch, M., Dimitrova, M., Ziane, K., Karganroudi, S.S., Dhouib, R., Ibrahim, H., Adda, M.: On predictive maintenance in industry 4.0: overview, models, and challenges. Appl. Sci. **12**(16), 8081 (2022). https://doi.org/10.3390/app12168081
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018). https://doi.org/10.1109/access.2018.2870052
3. Agrafiotis, I., Nurse, J.R.C., Goldsmith, M., Creese, S., Upton, D.: A taxonomy of cyber-harms: defining the impacts of cyber-attacks and understanding how they propagate. J. Cybersecur. **4**(1) (2018). https://doi.org/10.1093/cybsec/tyy006
4. Akowuah, F., Kong, F.: Real-time adaptive sensor attack detection in autonomous cyber-physical systems. In: 2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS). IEEE, Piscataway (2021). https://doi.org/10.1109/rtas52030.2021.00027
5. Akowuah, F., Prasad, R., Espinoza, C.O., Kong, F.: Recovery-by-learning: Restoring autonomous cyber-physical systems from sensor attacks. In: 2021 IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). IEEE, Piscataway (2021). https://doi.org/10.1109/rtcsa52859.2021.00015
6. Ala-Pietilä, P., Smuha, N.A.: A framework for global cooperation on artificial intelligence and its governance. In: Reflections on Artificial Intelligence for Humanity, pp. 237–265. Springer International Publishing, Cham (2021)
7. Alam, A.: Possibilities and apprehensions in the landscape of artificial intelligence in education. In: 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA). IEEE, Piscataway (2021). https://doi.org/10.1109/iccica52458.2021.9697272
8. Alkhalil, Z., Hewage, C., Nawaf, L., Khan, I.: Phishing attacks: a recent comprehensive study and a new anatomy. Front. Comput. Sci. **3** (2021). https://doi.org/10.3389/fcomp.2021.563060
9. Almeida, D., Shmarko, K., Lomas, E.: The ethics of facial recognition technologies, surveillance, and accountability in an age of artificial intelligence: a comparative analysis of US, EU, and UK regulatory frameworks. AI Ethics **2**(3), 377–387 (2021). https://doi.org/10.1007/s43681-021-00077-w
10. Alohali, M.A., Al-Wesabi, F.N., Hilal, A.M., Goel, S., Gupta, D., Khanna, A.: Artificial intelligence enabled intrusion detection systems for cognitive cyber-physical systems in industry 4.0 environment. Cognit. Neurodyn. **16**(5), 1045–1057 (2022). https://doi.org/10.1007/s11571-022-09780-8
11. Althobaiti, M.M., Kumar, K.P.M., Gupta, D., Kumar, S., Mansour, R.F.: An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems. Measurement **186**, 110145 (2021). https://doi.org/10.1016/j.measurement.2021.110145

12. Alwan, A.A., Ciupala, M.A., Brimicombe, A.J., Ghorashi, S.A., Baravalle, A., Falcarin, P.: Data quality challenges in large-scale cyber-physical systems: a systematic review. Inf. Syst. **105**, 101951 (2022). https://doi.org/10.1016/j.is.2021.101951
13. AlZubi, A.A., Al-Maitah, M., Alarifi, A.: Cyber-attack detection in healthcare using cyber-physical system and machine learning techniques. Soft Comput. **25**(18), 12319–12332 (2021). https://doi.org/10.1007/s00500-021-05926-8
14. An, N., Wang, X.: Legal protection of artificial intelligence data and algorithms from the perspective of internet of things resource sharing. Wirel. Commun. Mob. Comput. **2021**, 1–10 (2021). https://doi.org/10.1155/2021/8601425
15. Andraško, J., Mesarčík, M., Hamuľák, O.: The regulatory intersections between artificial intelligence, data protection and cyber security: challenges and opportunities for the EU legal framework. AI Soc. **36**, 623–636 (2021). https://doi.org/10.1007/s00146-020-01125-5
16. Anthi, E., Williams, L., Rhode, M., Burnap, P., Wedgbury, A.: Adversarial attacks on machine learning cybersecurity defences in industrial control systems. J. Inf. Secur. Appl. **58**, 102717 (2021). https://doi.org/10.1016/j.jisa.2020.102717
17. Arabo, A.: Cyber security challenges within the connected home ecosystem futures. Procedia Comput. Sci. **61**, 227–232 (2015). https://doi.org/10.1016/j.procs.2015.09.201
18. Bampoula, X., Siaterlis, G., Nikolakis, N., Alexopoulos, K.: A deep learning model for predictive maintenance in cyber-physical production systems using LSTM autoencoders. Sensors **21**(3), 972 (2021). https://doi.org/10.3390/s21030972
19. Barclay, I., Abramson, W.: Identifying roles, requirements and responsibilities in trustworthy AI systems. In: Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers. ACM, New York (2021). https://doi.org/10.1145/3460418.3479344
20. Bedué, P., Fritzsche, A.: Can we trust AI? An empirical investigation of trust requirements and guide to successful AI adoption. J. Enterp. Inf. Manag. **35**(2), 530–549 (2021). https://doi.org/10.1108/jeim-06-2020-0233
21. Bhattacharya, A., Ramachandran, T., Banik, S., Dowling, C.P., Bopardikar, S.D.: Automated adversary emulation for cyber-physical systems via reinforcement learning. In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, Piscataway (2020). https://doi.org/10.1109/isi49825.2020.9280521
22. Blair, J.R., Hall, A.O., Sobiesk, E.: Educating future multidisciplinary cybersecurity teams. Computer **52**(3), 58–66 (2019). https://doi.org/10.1109/mc.2018.2884190
23. Bokovnya, A.Y., Begishev, I.R., Khisamova, Z.I., Bikeev, I.I., Sidorenko, E.L., Bersei, D.D.: Pressing issues of unlawful application of artificial intelligence. Int. J. Criminol. Sociol. **9**, 1054–1057 (2020). https://doi.org/10.6000/1929-4409.2020.09.119
24. Broadbent, M.: What's ahead for a cooperative regulatory agenda on artificial intelligence? Center for Strategic and International Studies (CSIS) (2021)
25. Bulayenko, O., Quintais, J., Gervais, D.J., Poort, J.: AI music outputs: challenges to the copyright legal framework. SSRN Electronic Journal (2022). https://doi.org/10.2139/ssrn.4072806
26. Busuioc, M.: Accountable artificial intelligence: holding algorithms to account. Public Adm. Rev. **81**(5), 825–836 (2020). https://doi.org/10.1111/puar.13293
27. Calabrese, M., Cimmino, M., Fiume, F., Manfrin, M., Romeo, L., Ceccacci, S., Paolanti, M., Toscano, G., Ciandrini, G., Carrotta, A., Mengoni, M., Frontoni, E., Kapetis, D.: SOPHIA: an event-based IoT and machine learning architecture for predictive maintenance in industry 4.0. Information **11**(4), 202 (2020). https://doi.org/10.3390/info11040202
28. Carter, L., Liu, D., Cantrell, C.: Exploring the intersection of the digital divide and artificial intelligence: a hermeneutic literature review. AIS Trans. Human-Comput. Interact. **12**(4), 253–275 (2020). https://doi.org/10.17705/1thci.00138
29. Chalmers, D., MacKenzie, N.G., Carter, S.: Artificial intelligence and entrepreneurship: implications for venture creation in the fourth industrial revolution. Entrep. Theory Pract. **45**(5), 1028–1053 (2020). https://doi.org/10.1177/1042258720934581

30. Chin, Y.C., Zhao, J.: Governing cross-border data flows: international trade agreements and their limits. Laws **11**(4), 63 (2022). https://doi.org/10.3390/laws11040063
31. Cho, S., May, G., Tourkogiorgis, I., Perez, R., Lazaro, O., de la Maza, B., Kiritsis, D.: A hybrid machine learning approach for predictive maintenance in smart factories of the future. In: Advances in Production Management Systems. Smart Manufacturing for Industry 4.0, pp. 311–317. Springer International Publishing, Cham (2018)
32. Clarke, R.: Regulatory alternatives for AI. Comput. Law Secur. Rev. **35**(4), 398–409 (2019). https://doi.org/10.1016/j.clsr.2019.04.008
33. Coeckelbergh, M.: Artificial intelligence: some ethical issues and regulatory challenges. Technol. Regulat. **2019**(2019) (2019). https://doi.org/10.26116/TECHREG.2019.003
34. Corallo, A., Lazoi, M., Lezzi, M.: Cybersecurity in the context of industry 4.0: a structured classification of critical assets and business impacts. Comput. Ind. **114**, 103165 (2020). https://doi.org/10.1016/j.compind.2019.103165
35. Dai, W., Dubinin, V.N., Christensen, J.H., Vyatkin, V., Guan, X.: Toward self-manageable and adaptive industrial cyber-physical systems with knowledge-driven autonomic service management. IEEE Trans. Ind. Inf. **13**(2), 725–736 (2017). https://doi.org/10.1109/tii.2016.2595401
36. Deng, W.H., Yildirim, N., Chang, M., Eslami, M., Holstein, K., Madaio, M.: Investigating practices and opportunities for cross-functional collaboration around AI fairness in industry practice. In: 2023 ACM Conference on Fairness, Accountability, and Transparency. ACM, New York (2023). https://doi.org/10.1145/3593013.3594037
37. Dwivedi, Y.K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Ilavarasan, P.V., Janssen, M., Jones, P., Kar, A.K., Kizgin, H., Kronemann, B., Lal, B., Lucini, B., Medaglia, R., Meunier-FitzHugh, K.L., Meunier-FitzHugh, L.C.L., Misra, S., Mogaji, E., Sharma, S.K., Singh, J.B., Raghavan, V., Raman, R., Rana, N.P., Samothrakis, S., Spencer, J., Tamilmani, K., Tubadji, A., Walton, P., Williams, M.D.: Artificial intelligence (AI): multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. Int. J. Inf. Manag. **57**, 101994 (2021). https://doi.org/10.1016/j.ijinfomgt.2019.08.002
38. Eaton, C., Ramkumar, A.: Colonial pipeline shutdown: is there a gas shortage and when will the pipeline be fixed? Wall Street J. (2021). https://www.wsj.com/articles/colonial-pipeline-cyberattack-hack-11620668583
39. Ehsan, U., Liao, Q.V., Muller, M., Riedl, M.O., Weisz, J.D.: Expanding explainability: Towards social transparency in ai systems. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, New York (2021). https://doi.org/10.1145/3411764.3445188
40. Elgendi, I., Hossain, M.F., Jamalipour, A., Munasinghe, K.S.: Protecting cyber physical systems using a learned MAPE-k model. IEEE Access **7**, 90954–90963 (2019). https://doi.org/10.1109/access.2019.2927037
41. Elnaggar, M., Bezzo, N.: An IRL approach for cyber-physical attack intention prediction and recovery. In: 2018 Annual American Control Conference (ACC). IEEE, Piscataway (2018). https://doi.org/10.23919/acc.2018.8430922
42. European Commision: Draft ethics guidelines for trustworthy AI (2018). https://ec.europa.eu/digital-single-market/en/news/draft-ethics-guidelines-trustworthy-ai
43. European Commission: Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts (2021). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52021PC0206
44. European Union Agency for Cybersecurity.: Identifying emerging cybersecurity threats and challenges for 2030. Publications Office (2023). https://doi.org/10.2824/117542
45. Felzmann, H., Villaronga, E.F., Lutz, C., Tamò-Larrieux, A.: Transparency you can trust: transparency requirements for artificial intelligence between legal norms and contextual concerns. Big Data Soc. **6**(1), 205395171986054 (2019). https://doi.org/10.1177/2053951719860542

46. Felzmann, H., Fosch-Villaronga, E., Lutz, C., Tamò-Larrieux, A.: Towards transparency by design for artificial intelligence. Sci. Eng. Ethics **26**(6), 3333–3361 (2020). https://doi.org/10.1007/s11948-020-00276-4
47. Ferrer, X., van Nuenen, T., Such, J.M., Cote, M., Criado, N.: Bias and discrimination in AI: a cross-disciplinary perspective. IEEE Technol. Soc. Mag. **40**(2), 72–80 (2021). https://doi.org/10.1109/mts.2021.3056293
48. Floridi, L., Cowls, J., King, T.C., Taddeo, M.: How to design AI for social good: Seven essential factors. Sci. Eng. Ethics **26**(3), 1771–1796 (2020). https://doi.org/10.1007/s11948-020-00213-5
49. Fontes, C., Hohma, E., Corrigan, C.C., Lütge, C.: AI-powered public surveillance systems: why we (might) need them and how we want them. Technol. Soc. **71**, 102137 (2022). https://doi.org/10.1016/j.techsoc.2022.102137
50. Frank, M.R., Autor, D., Bessen, J.E., Brynjolfsson, E., Cebrian, M., Deming, D.J., Feldman, M., Groh, M., Lobo, J., Moro, E., Wang, D., Youn, H., Rahwan, I.: Toward understanding the impact of artificial intelligence on labor. Proc. Natl. Acad. Sci. **116**(14), 6531–6539 (2019). https://doi.org/10.1073/pnas.1900949116
51. Gabriel, I.: Artificial intelligence, values, and alignment. Minds Mach. **30**(3), 411–437 (2020). https://doi.org/10.1007/s11023-020-09539-2
52. Ganesh, P., Lou, X., Chen, Y., Tan, R., Yau, D.K.Y., Chen, D., Winslett, M.: Learning-based simultaneous detection and characterization of time delay attack in cyber-physical systems. IEEE Trans. Smart Grid **12**(4), 3581–3593 (2021). https://doi.org/10.1109/tsg.2021.3058682
53. Gerke, S., Minssen, T., Cohen, G.: Ethical and legal challenges of artificial intelligence-driven healthcare. In: Artificial Intelligence in Healthcare, pp. 295–336. Elsevier, Amsterdam (2020). https://doi.org/10.1016/b978-0-12-818438-7.00012-5
54. Gill, S.S., Xu, M., Ottaviani, C., Patros, P., Bahsoon, R., Shaghaghi, A., Golec, M., Stankovski, V., Wu, H., Abraham, A., Singh, M., Mehta, H., Ghosh, S.K., Baker, T., Parlikad, A.K., Lutfiyya, H., Kanhere, S.S., Sakellariou, R., Dustdar, S., Rana, O., Brandic, I., Uhlig, S.: AI for next generation computing: emerging trends and future directions. Int. Things **19**, 100514 (2022). https://doi.org/10.1016/j.iot.2022.100514
55. Gomez, A.L.P., Maimo, L.F., Celdran, A.H., Clemente, F.J.G., Sarmiento, C.C., Masa, C.J.D.C., Nistal, R.M.: On the generation of anomaly detection datasets in industrial control systems. IEEE Access **7**, 177460–177473 (2019). https://doi.org/10.1109/access.2019.2958284
56. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: XAI—explainable artificial intelligence. Sci. Rob. **4**(37) (2019). https://doi.org/10.1126/scirobotics.aay7120
57. Guzman, N.H.C., Wied, M., Kozine, I., Lundteigen, M.A.: Conceptualizing the key features of cyber-physical systems in a multi-layered representation for safety and security analysis. Syst. Eng. **23**(2), 189–210 (2019). https://doi.org/10.1002/sys.21509
58. Guzman, N.H.C., Kozine, I., Lundteigen, M.A.: An integrated safety and security analysis for cyber-physical harm scenarios. Safety Sci. **144**, 105458 (2021). https://doi.org/10.1016/j.ssci.2021.105458
59. Halisdemir, M.E., Karacan, H., Pihelgas, M., Lepik, T., Cho, S.: Data quality problem in AI-based network intrusion detection systems studies and a solution proposal. In: 2022 14th International Conference on Cyber Conflict: Keep Moving! (CyCon). IEEE, Piscataway (2022). https://doi.org/10.23919/cycon55549.2022.9811014
60. Hamon, R., Junklewitz, H., Sanchez, I., Malgieri, G., Hert, P.D.: Bridging the gap between AI and explainability in the GDPR: towards trustworthiness-by-design in automated decision-making. IEEE Comput. Intell. Mag. **17**(1), 72–85 (2022). https://doi.org/10.1109/mci.2021.3129960
61. Hao, W., Yang, T., Yang, Q.: Hybrid statistical-machine learning for real-time anomaly detection in industrial cyber–physical systems. IEEE Trans. Automat. Sci. Eng. **20**(1), 32–46 (2023). https://doi.org/10.1109/tase.2021.3073396
62. Harris, M.A., Patten, K.P.: Mobile device security considerations for small- and medium-sized enterprise business mobility. Inf. Manag. Comput. Secur. **22**(1), 97–114 (2014). https://doi.org/10.1108/imcs-03-2013-0019

63. Hasan, M., Islam, M.M., Zarif, M.I.I., Hashem, M.: Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Int. Things **7**, 100059 (2019). https://doi.org/10.1016/j.iot.2019.100059
64. Holmes, W., Porayska-Pomsta, K., Holstein, K., Sutherland, E., Baker, T., Shum, S.B., Santos, O.C., Rodrigo, M.T., Cukurova, M., Bittencourt, I.I., Koedinger, K.R.: Ethics of AI in education: towards a community-wide framework. Int. J. Artif. Intell. Educat. **32**(3), 504–526 (2021). https://doi.org/10.1007/s40593-021-00239-1
65. Innovation Eye: Artificial intelligence industry in the UK landscape overview 2021: Companies, investors, influencers and trends (2021). https://analytics.dkv.global/AI-in-UK-2021/Report.pdf
66. Jamshidi, P., Camara, J., Schmerl, B., Kaestner, C., Garlan, D.: Machine learning meets quantitative planning: Enabling self-adaptation in autonomous robots. In: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS). IEEE, Piscataway (2019). https://doi.org/10.1109/seams.2019.00015
67. Janssen, M., Brous, P., Estevez, E., Barbosa, L.S., Janowski, T.: Data governance: organizing data for trustworthy artificial intelligence. Govern. Inf. Quart. **37**(3), 101493 (2020). https://doi.org/10.1016/j.giq.2020.101493
68. Kanimozhi, V., Jacob, T.P.: Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In: 2019 International Conference on Communication and Signal Processing (ICCSP). IEEE, Piscataway (2019). https://doi.org/10.1109/iccsp.2019.8698029
69. Kelley, K.H., Fontanetta, L.M., Heintzman, M., Pereira, N.: Artificial intelligence: implications for social inflation and insurance. Risk Manag. Insur. Rev. **21**(3), 373–387 (2018). https://doi.org/10.1111/rmir.12111
70. Keskinbora, K.H.: Medical ethics considerations on artificial intelligence. J. Clin. Neurosci. **64**, 277–282 (2019). https://doi.org/10.1016/j.jocn.2019.03.001
71. Khan, A.A., Beg, O.A., Alamaniotis, M., Ahmed, S.: Intelligent anomaly identification in cyber-physical inverter-based systems. Electr. Power Syst. Res. **193**, 107024 (2021). https://doi.org/10.1016/j.epsr.2021.107024
72. Kholidy, H.A.: Autonomous mitigation of cyber risks in the cyber–physical systems. Future Gener. Comput. Syst. **115**, 171–187 (2021). https://doi.org/10.1016/j.future.2020.09.002
73. Kimani, K., Oduol, V., Langat, K.: Cyber security challenges for IoT-based smart grid networks. Int. J. Critic. Infrastruct. Protect. **25**, 36–49 (2019). https://doi.org/10.1016/j.ijcip.2019.01.001
74. Kingston, J.K.C.: Artificial intelligence and legal liability. In: Research and Development in Intelligent Systems XXXIII, pp. 269–279. Springer International Publishing, Cham (2016)
75. Kitsara, I.: Artificial intelligence and the digital divide: From an innovation perspective. In: Platforms and Artificial Intelligence, pp. 245–265. Springer International Publishing, Cham (2022)
76. Kreimel, P., Eigner, O., Tavolato, P.: Anomaly-based detection and classification of attacks in cyber-physical systems. In: Proceedings of the 12th International Conference on Availability, Reliability and Security. ACM, New York (2017). https://doi.org/10.1145/3098954.3103155
77. Kuziemski, M., Misuraca, G.: AI governance in the public sector: Three tales from the frontiers of automated decision-making in democratic settings. Telecommun. Policy **44**(6), 101976 (2020). https://doi.org/10.1016/j.telpol.2020.101976
78. Larsson, S.: On the governance of artificial intelligence through ethics guidelines. Asian J. Law Soc. **7**(3), 437–451 (2020). https://doi.org/10.1017/als.2020.19
79. Laso, P.M., Brosset, D., Puentes, J.: Dataset of anomalies and malicious acts in a cyber-physical subsystem. Data Brief **14**, 186–191 (2017). https://doi.org/10.1016/j.dib.2017.07.038
80. Laupichler, M.C., Aster, A., Schirch, J., Raupach, T.: Artificial intelligence literacy in higher and adult education: a scoping literature review. Comput. Educ. Artif. Intell. **3**, 100101 (2022). https://doi.org/10.1016/j.caeai.2022.100101

81. Lee, J.: Access to finance for artificial intelligence regulation in the financial services industry. Eur. Business Organiz. Law Rev. **21**(4), 731–757 (2020). https://doi.org/10.1007/s40804-020-00200-0
82. Lee, M.F.R., Chien, T.W.: Artificial intelligence and internet of things for robotic disaster response. In: 2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS). IEEE, Piscataway (2020). https://doi.org/10.1109/aris50834.2020.9205794
83. Li, B., Wu, Y., Song, J., Lu, R., Li, T., Zhao, L.: DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Trans. Ind. Inf. **17**(8), 5615–5624 (2021). https://doi.org/10.1109/tii.2020.3023430
84. Li, Y., Yan, J., Naili, M.: Deep reinforcement learning for penetration testing of cyber-physical attacks in the smart grid. In: 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, Piscataway (2022). https://doi.org/10.1109/ijcnn55064.2022.9892584
85. Liu, Q., Hagenmeyer, V., Keller, H.B.: A review of rule learning-based intrusion detection systems and their prospects in smart grids. IEEE Access **9**, 57542–57564 (2021). https://doi.org/10.1109/access.2021.3071263
86. Long, D., Magerko, B.: What is AI literacy? Competencies and design considerations. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. ACM, New York (2020). https://doi.org/10.1145/3313831.3376727
87. Luo, X., Wu, Y., Xiao, X., Ooi, B.C.: Feature inference attack on model predictions in vertical federated learning. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, Piscataway (2021). https://doi.org/10.1109/icde51399.2021.00023
88. Luo, Y., Xiao, Y., Cheng, L., Peng, G., Yao, D.D.: Deep learning-based anomaly detection in cyber-physical systems. ACM Comput. Surv. **54**(5), 1–36 (2021). https://doi.org/10.1145/3453155
89. Lutz, C.: Digital inequalities in the age of artificial intelligence and big data. Human Behav. Emerg. Technol. **1**(2), 141–148 (2019). https://doi.org/10.1002/hbe2.140
90. Lv, Z., Chen, D., Lou, R., Alazab, A.: Artificial intelligence for securing industrial-based cyber–physical systems. Future Gener. Comput. Syst. **117**, 291–298 (2021). https://doi.org/10.1016/j.future.2020.12.001
91. Madaio, M.A., Stark, L., Vaughan, J.W., Wallach, H.: Co-designing checklists to understand organizational challenges and opportunities around fairness in AI. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. ACM, New York (2020). https://doi.org/10.1145/3313831.3376445
92. Maedche, A., Legner, C., Benlian, A., Berger, B., Gimpel, H., Hess, T., Hinz, O., Morana, S., Söllner, M.: AI-based digital assistants. Business Inf. Syst. Eng. **61**(4), 535–544 (2019). https://doi.org/10.1007/s12599-019-00600-8
93. Maleh, Y.: Machine learning techniques for IoT intrusions detection in aerospace cyber-physical systems. In: Studies in Computational Intelligence, pp. 205–232. Springer International Publishing, Cham (2019)
94. Mansour, R.F.: Artificial intelligence based optimization with deep learning model for blockchain enabled intrusion detection in CPS environment. Sci. Rep. **12**(1) (2022). https://doi.org/10.1038/s41598-022-17043-z
95. Marda, V.: Artificial intelligence policy in india: a framework for engaging the limits of data-driven decision-making. Philos. Trans. Roy. Soc. A Math. Phys. Eng. Sci. **376**(2133), 20180087 (2018). https://doi.org/10.1098/rsta.2018.0087
96. Marino, D.L., Wickramasinghe, C.S., Singh, V.K., Gentle, J., Rieger, C., Manic, M.: The virtualized cyber-physical testbed for machine learning anomaly detection: a wind powered grid case study. IEEE Access **9**, 159475–159494 (2021). https://doi.org/10.1109/access.2021.3127169
97. Matsuda, W., Fujimoto, M., Aoyama, T., Mitsunaga, T.: Cyber security risk assessment on industry 4.0 using ICS testbed with AI and cloud. In: 2019 IEEE Conference on Application, Information and Network Security (AINS). IEEE, Piscataway (2019). https://doi.org/10.1109/ains47559.2019.8968698

98. Matus, K.J., Veale, M.: Certification systems for machine learning: lessons from sustainability. Regul. Govern. **16**(1), 177–196 (2021). https://doi.org/10.1111/rego.12417
99. McKinnel, D.R., Dargahi, T., Dehghantanha, A., Choo, K.K.R.: A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment. Comput. Electr. Eng. **75**, 175–188 (2019). https://doi.org/10.1016/j.compeleceng.2019.02.022
100. Meesublak, K., Klinsukont, T.: A cyber-physical system approach for predictive maintenance. In: 2020 IEEE International Conference on Smart Internet of Things (SmartIoT). IEEE, Piscataway (2020). https://doi.org/10.1109/smartiot49966.2020.00061
101. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6), 1–35 (2021). https://doi.org/10.1145/3457607
102. Mittelstadt, B.: Principles alone cannot guarantee ethical AI. Nat. Mach. Intell. **1**(11), 501–507 (2019). https://doi.org/10.1038/s42256-019-0114-4
103. Morozova, G.A., Kuznetsov, V.P., Kozlova, E.P., Zaitseva, S.A., Andryashina, N.S.: The impact of artificial intelligence on the socio-economic development of society in modern conditions. In: Current Problems and Ways of Industry Development: Equipment and Technologies, pp. 406–414. Springer International Publishing, Cham (2021)
104. Muscanell, N.L., Guadagno, R.E., Murphy, S.: Weapons of influence misused: a social influence analysis of why people fall prey to internet scams. Soc. Person. Psychol. Compass **8**(7), 388–396 (2014). https://doi.org/10.1111/spc3.12115
105. Naik, N., Hameed, B.M.Z., Shetty, D.K., Swain, D., Shah, M., Paul, R., Aggarwal, K., Ibrahim, S., Patil, V., Smriti, K., Shetty, S., Rai, B.P., Chlosta, P., Somani, B.K.: Legal and ethical consideration in artificial intelligence in healthcare: who takes responsibility? Front. Surgery **9** (2022). https://doi.org/10.3389/fsurg.2022.862322
106. Ng, D.T.K., Leung, J.K.L., Chu, S.K.W., Qiao, M.S.: Conceptualizing AI literacy: an exploratory review. Comput. Educat. Artif. Intell. **2**, 100041 (2021). https://doi.org/10.1016/j.caeai.2021.100041
107. Nozomi Networks: Regional power operator improves ics cybersecurity and operational efficiency. Technical Report, Nozomi Networks and Vermont Electric Coop (2021). https://www.nozominetworks.com/downloads/US/Nozomi-Networks-VEC-Case-Study.pdf
108. Nweke, L.O.: Using the cia and aaa models to explain cybersecurity activities. PM World J. **6**, 1–3 (2017)
109. Nweke, L.O.: A survey of specification-based intrusion detection techniques for cyber-physical systems. Int. J. Adv. Comput. Sci. Appl. **12**(5) (2021). https://doi.org/10.14569/ijacsa.2021.0120506
110. Nweke, L.O.: Using formal methods for modelling cyber-physical systems secrity. Ph.D. Thesis, Norwegian University of Science and Technology (NTNU) (2022)
111. Nweke, L.O., Wolthusen, S.D.: A review of asset-centric threat modelling approaches. Int. J. Adv. Comput. Sci. Appl. **11**(2) (2020). https://doi.org/10.14569/ijacsa.2020.0110201
112. Nweke, L.O., Wolthusen, S.: Legal issues related to cyber threat information sharing among private entities for critical infrastructure protection. In: 2020 12th International Conference on Cyber Conflict (CyCon). IEEE, Piscataway (2020). https://doi.org/10.23919/cycon49761.2020.9131721
113. Nweke, L.O., Wolthusen, S.D.: A holistic approach for enhancing critical infrastructure protection: Research agenda. In: International Conference on Emerging Applications and Technologies for Industry 4.0 (EATI'2020), pp. 220–228. Springer International Publishing, Cham (2021)
114. Nweke, L.O., Weldehawaryat, G.K., Wolthusen, S.D.: Threat modelling of cyber-physical systems using an applied pi-calculus. Int. J. Crit. Infrastruct. Protect. **35**, 100466 (2021). https://doi.org/10.1016/j.ijcip.2021.100466
115. ÓhÉigeartaigh, S.S., Whittlestone, J., Liu, Y., Zeng, Y., Liu, Z.: Overcoming barriers to cross-cultural cooperation in AI ethics and governance. Philos. Technol. **33**(4), 571–593 (2020). https://doi.org/10.1007/s13347-020-00402-x

116. Oliveira, N., Sousa, N., Oliveira, J., Praca, I.: Anomaly detection in cyber-physical systems: Reconstruction of a prediction error feature space. In: 2021 14th International Conference on Security of Information and Networks (SIN). IEEE, Piscataway (2021). https://doi.org/10.1109/sin54109.2021.9699339
117. Olowononi, F.O., Rawat, D.B., Liu, C.: Resilient machine learning for networked cyber physical systems: A survey for machine learning security to securing machine learning for CPS. IEEE Commun. Surv. Tutor. **23**(1), 524–552 (2021). https://doi.org/10.1109/comst.2020.3036778
118. O'Sullivan, S., Nevejans, N., Allen, C., Blyth, A., Leonard, S., Pagallo, U., Holzinger, K., Holzinger, A., Sajid, M.I., Ashrafian, H.: Legal, regulatory, and ethical frameworks for development of standards in artificial intelligence (AI) and autonomous robotic surgery. Int. J. Med. Rob. Comput. Assist. Surgery **15**(1), e1968 (2019). https://doi.org/10.1002/rcs.1968
119. Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., Loncarski, J.: Machine learning approach for predictive maintenance in industry 4.0. In: 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA). IEEE, Piscataway (2018). https://doi.org/10.1109/mesa.2018.8449150
120. Paredes, C.M., Martínez-Castro, D., Ibarra-Junquera, V., González-Potes, A.: Detection and isolation of DoS and integrity cyber attacks in cyber-physical systems with a neural network-based architecture. Electronics **10**(18), 2238 (2021). https://doi.org/10.3390/electronics10182238
121. Peng, S.Y.: Cybersecurity threats and the WTO national security exceptions. J. Int. Econ. Law **18**(2), 449–478 (2015). https://doi.org/10.1093/jiel/jgv025
122. Peres, R.S., Jia, X., Lee, J., Sun, K., Colombo, A.W., Barata, J.: Industrial artificial intelligence in industry 4.0 - systematic review, challenges and outlook. IEEE Access **8**, 220121–220139 (2020). https://doi.org/10.1109/access.2020.3042874
123. Pot, M., Kieusseyan, N., Prainsack, B.: Not all biases are bad: equitable and inequitable biases in machine learning and radiology. Insights Imag. **12**(1) (2021). https://doi.org/10.1186/s13244-020-00955-7
124. Qiu, S., Liu, Q., Zhou, S., Wu, C.: Review of artificial intelligence adversarial attack and defense technologies. Appl. Sci. **9**(5), 909 (2019). https://doi.org/10.3390/app9050909
125. Qiu, H., Qiu, M., Liu, M., Memmi, G.: Secure health data sharing for medical cyber-physical systems for the healthcare 4.0. IEEE J. Biomed. Health Inf. **24**(9), 2499–2505 (2020). https://doi.org/10.1109/jbhi.2020.2973467
126. Raji, I.D., Smart, A., White, R.N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., Barnes, P.: Closing the AI accountability gap. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. ACM, New York (2020). https://doi.org/10.1145/3351095.3372873
127. Ramasamy, L.K., Khan, F., Shah, M., Prasad, B.V.V.S., Iwendi, C., Biamba, C.: Secure smart wearable computing through artificial intelligence-enabled internet of things and cyber-physical systems for health monitoring. Sensors **22**(3), 1076 (2022). https://doi.org/10.3390/s22031076
128. Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning (2016). https://doi.org/10.48550/ARXIV.1606.05386
129. Rouzbahani, H.M., Karimipour, H., Rahimnejad, A., Dehghantanha, A., Srivastava, G.: Anomaly detection in cyber-physical systems using machine learning. In: Handbook of Big Data Privacy, pp. 219–235. Springer International Publishing, Cham (2020)
130. Saha, T., Aaraj, N., Ajjarapu, N., Jha, N.K.: SHARKS: Smart hacking approaches for RisK scanning in internet-of-things and cyber-physical systems based on machine learning. IEEE Trans. Emerg. Top. Comput. **10**, 1–1 (2021). https://doi.org/10.1109/tetc.2021.3050733
131. Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., Aroyo, L.M.: "everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. ACM, New York (2021). https://doi.org/10.1145/3411764.3445518

132. Sanger, D.E., Krauss, C., Perlroth, N.: Cyberat-tack forces a shutdown of a top U.S. pipeline (2021). https://www.nytimes.com/2021/05/08/us/politics/cyberattack-colonial-pipeline.html
133. Sayghe, A., Zhao, J., Konstantinou, C.: Evasion attacks with adversarial deep learning against power system state estimation. In: 2020 IEEE Power and Energy Society General Meeting (PESGM). IEEE, Piscataway (2020). https://doi.org/10.1109/pesgm41954.2020.9281719
134. Sen, R., Heim, G., Zhu, Q.: Artificial intelligence and machine learning in cybersecurity: applications, challenges, and opportunities for MIS academics. Commun. Assoc. Inf. Syst. **51**(1), 179–209 (2022). https://doi.org/10.17705/1cais.05109
135. Sengan, S., Subramaniyaswamy, V., Indragandhi, V., Velayutham, P., Ravi, L.: Detection of false data cyber-attacks for the assessment of security in smart grid using deep learning. Comput. Electr. Eng. **93**, 107211 (2021). https://doi.org/10.1016/j.compeleceng.2021.107211
136. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy. SCITEPRESS - Science and Technology Publications (2018). https://doi.org/10.5220/0006639801080116
137. Shin, D.: The effects of explainability and causability on perception, trust, and acceptance: implications for explainable AI. Int. J. Human-Comput. Stud. **146**, 102551 (2021). https://doi.org/10.1016/j.ijhcs.2020.102551
138. Shin, H.K., Lee, W., Yun, J.H., Kim, H.: Implementation of programmable CPS testbed for anomaly detection. In: CSET'19: Proceedings of the 12th USENIX Conference on Cyber Security Experimentation and Test (2019)
139. Shin, H.K., Lee, W., Yun, J.H., Kim, H.: Hai 1.0: Hil-based augmented ICS security dataset. In: CSET'20: Proceedings of the 13th USENIX Conference on Cyber Security Experimentation and Test (2020)
140. Siau, K., Wang, W.: Artificial intelligence (AI) ethics. J. Database Manag. **31**(2), 74–87 (2020). https://doi.org/10.4018/jdm.2020040105
141. Song, L., Wang, L., Wu, J., Liang, J., Liu, Z.: Integrating physics and data driven cyber-physical system for condition monitoring of critical transmission components in smart production line. Appl. Sci. **11**(19), 8967 (2021). https://doi.org/10.3390/app11198967
142. Sreedevi, A., Harshitha, T.N., Sugumaran, V., Shankar, P.: Application of cognitive computing in healthcare, cybersecurity, big data and IoT: a literature review. Inf. Process. Manag. **59**(2), 102888 (2022). https://doi.org/10.1016/j.ipm.2022.102888
143. Staves, A., Anderson, T., Balderstone, H., Green, B., Gouglidis, A., Hutchison, D.: A cyber incident response and recovery framework to support operators of industrial control systems. Int. J. Crit. Infrastruct. Protect. **37**, 100505 (2022). https://doi.org/10.1016/j.ijcip.2021.100505
144. Suhail, S., Malik, S.U.R., Jurdak, R., Hussain, R., Matulevičius, R., Svetinovic, D.: Towards situational aware cyber-physical systems: a security-enhancing use case of blockchain-based digital twins. Comput. Ind. **141**, 103699 (2022). https://doi.org/10.1016/j.compind.2022.103699
145. Tamers, S.L., Streit, J., Pana-Cryan, R., Ray, T., Syron, L., Flynn, M.A., Castillo, D., Roth, G., Geraci, C., Guerin, R., Schulte, P., Henn, S., Chang, C.C., Felknor, S., Howard, J.: Envisioning the future of work to safeguard the safety, health, and well-being of the workforce: a perspective from the CDC's national institute for occupational safety and health. Amer. J. Ind. Med. **63**(12), 1065–1084 (2020). https://doi.org/10.1002/ajim.23183
146. Teng, S.Y., Touš, M., Leong, W.D., How, B.S., Lam, H.L., Máša, V.: Recent advances on industrial data-driven energy savings: digital twins and infrastructures. Renew. Sustain. Energy Rev. **135**, 110208 (2021). https://doi.org/10.1016/j.rser.2020.110208
147. Theodorou, A., Dignum, V.: Towards ethical and socio-legal governance in AI. Nat. Mach. Intell. **2**(1), 10–12 (2020). https://doi.org/10.1038/s42256-019-0136-y
148. Tian, J., Wang, B., Li, J., Wang, Z., Ma, B., Ozay, M.: Exploring targeted and stealthy false data injection attacks via adversarial machine learning. IEEE Int. Things J. **9**(15), 14116–14125 (2022). https://doi.org/10.1109/jiot.2022.3147040

149. Tolpegin, V., Truex, S., Gursoy, M.E., Liu, L.: Data poisoning attacks against federated learning systems. In: Computer Security – ESORICS 2020, pp. 480–501. Springer International Publishing, Cham (2020)
150. Tomašev, N., Cornebise, J., Hutter, F., Mohamed, S., Picciariello, A., Connelly, B., Belgrave, D.C.M., Ezer, D., van der Haert, F.C., Mugisha, F., Abila, G., Arai, H., Almiraat, H., Proskurnia, J., Snyder, K., Otake-Matsuura, M., Othman, M., Glasmachers, T., de Wever, W., Teh, Y.W., Khan, M.E., Winne, R.D., Schaul, T., Clopath, C.: AI for social good: unlocking the opportunity for positive impact. Nat. Commun. **11**(1) (2020). https://doi.org/10.1038/s41467-020-15871-z
151. Trattner, C., Jannach, D., Motta, E., Meijer, I.C., Diakopoulos, N., Elahi, M., Opdahl, A.L., Tessem, B., Borch, N., Fjeld, M., Øvrelid, L., Smedt, K.D., Moe, H.: Responsible media technology and AI: challenges and research directions. AI Ethics **2**(4), 585–594 (2021). https://doi.org/10.1007/s43681-021-00126-4
152. Ulnicane, I., Eke, D.O., Knight, W., Ogoh, G., Stahl, B.C.: Good governance as a response to discontents? déjà vu, or lessons for AI from other emerging technologies. Interdiscipl. Sci. Rev. **46**(1–2), 71–93 (2021). https://doi.org/10.1080/03080188.2020.1840220
153. Vasse'i, R.M.: The ethical guidelines for trustworthy AI – a procrastination of effective law enforcement. Comput. Law Rev. Int. **20**(5), 129–136 (2019). https://doi.org/10.9785/cri-2019-200502
154. Wachter, S., Mittelstadt, B., Russell, C.: Why fairness cannot be automated: bridging the gap between EU non-discrimination law and AI. Comput. Law Secur. Rev. **41**, 105567 (2021). https://doi.org/10.1016/j.clsr.2021.105567
155. Wan, B., Xu, C., Mahapatra, R.P., Selvaraj, P.: Understanding the cyber-physical system in international stadiums for security in the network from cyber-attacks and adversaries using AI. Wirel. Person. Commun. **127**(2), 1207–1224 (2021). https://doi.org/10.1007/s11277-021-08573-2
156. Wang, P.: On defining artificial intelligence. J. Artif. General Intell. **10**(2), 1–37 (2019). https://doi.org/10.2478/jagi-2019-0002
157. Werder, K., Ramesh, B., Zhang, R.S.: Establishing data provenance for responsible artificial intelligence systems. ACM Trans. Manag. Inf. Syst. **13**(2), 1–23 (2022). https://doi.org/10.1145/3503488
158. Wickramasinghe, C.S., Marino, D.L., Amarasinghe, K., Manic, M.: Generalization of deep learning for cyber-physical system security: A survey. In: IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society. IEEE, Piscataway (2018). https://doi.org/10.1109/iecon.2018.8591773
159. Wilson, C.: Public engagement and AI: a values analysis of national strategies. Governm. Inf. Quart. **39**(1), 101652 (2022). https://doi.org/10.1016/j.giq.2021.101652
160. Winfield, A.F.T., Jirotka, M.: Ethical governance is essential to building trust in robotics and artificial intelligence systems. Philos. Trans. Roy. Soc. A Math. Phys. Eng. Sci. **376**(2133), 20180085 (2018). https://doi.org/10.1098/rsta.2018.0085
161. Zarina, I.K., Ildar R.B., Elina L.S.: Artificial intelligence and problems of ensuring cyber security. Int. J. Cyber Criminol. (2020). https://doi.org/10.5281/ZENODO.3709267
162. Zwilling, M., Klien, G., Lesjak, D., Wiechetek, Ł., Cetin, F., Basim, H.N.: Cyber security awareness, knowledge and behavior: a comparative study. J. Comput. Inf. Syst. **62**(1), 82–97 (2020). https://doi.org/10.1080/08874417.2020.1712269

Artificial Intelligence Working to Secure Small Enterprises

Kai Rasmus

1 Introduction

Small- and medium-sized enterprises (SMEs) represent a majority of all enterprises globally and 92% of all enterprises in Finland [1]. An important feature of SMEs is their limited number of resources available for information and communication technology (ICT) administration and cyber security [2]. Recent research has shown that especially situational awareness and the prevalence of bad security practices are the main areas in which more research is needed. Bad security practices can arise from a belief that the small size of an SME secures it from attacks [2]. The size of an enterprise is not a security control against cyber-attack. In 2014, a study of Internet threats found that 30% of attacks specifically targeted SMEs [3].

Malware technology is also developing [4]. New and emerging AI-based malware is making signature-based, behavioural and heuristic antimalware software almost obsolete. Malware tools are being made available online that are claimed to be fully undetectable by present-day antimalware software [5]. The amount of sophisticated phishing scams arriving over the network, or via e-mail, is increasing, and these are becoming harder for users to detect [6].

The modern-day SME is also not necessarily located in one physical place, with employees possibly telecommuting and important resources being located on third-party external servers on the Internet (i.e. 'the cloud'). This type of distributed office scenario has many more attack vectors and threats than a traditional office that is totally confined in a local network behind one corporate firewall. Telecommuting can also be forced on SME employees due to unforeseen events such as global pandemics [7].

K. Rasmus (✉)
Luode Consulting Oy, Jyväskylä, Finland
e-mail: kai.rasmus@luode.fi

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_8

These developments in the threat and business environment are leaving SMEs vulnerable. The traditional way of securing an enterprise is to have a set of tools or a security operations centre (SOC) that monitors signals emanating from security sensors located around the network. Rulesets have been predefined to tell the sensors when to trigger. These rules cannot be too narrow because then real problems could be missed. This means that the sensors generally generate a lot of messages including many false positives. The SOC may miss attacks in this mass of data or they may notice them too late [8]. In many cases the results can also be hard to interpret. Even as more tools are being introduced, research has found that SMEs need information more on vulnerabilities than implementation of tools [9].

The analysis of this large amount of cyber security data is therefore crucial but requires a lot of human resources to accomplish properly. There exists a role for an AI system in analysing this data based on well-researched scientific techniques, especially in the fields of machine learning and deep learning. AI systems are well suited to identifying patterns in the data, thus reducing the need for human intervention [10]. Taking this into account, AI-based cyber security tools and methods could help in countering the new and emerging threats facing SMEs. These tools have been developed for different purposes and work either separately or together with existing antimalware, firewalls and intrusion detection systems. In these roles, the tools can be located in different parts of the cyber security space.

Cyber security management becomes most effective when it is implemented using a framework. In this way the most important aspects are taken into account, and it becomes easier to become standards-compliant. One extensively utilized example is the National Institute of Standards and Technology (NIST) framework which has been developed around five core functions: identify, protect, detect, respond and recover [11]. The controls in NIST can be mapped onto those of the ISO27001 information security standard [12] but introduces them in a more user-friendly way. This framework is frequently updated to take into account the changing technology environment. For example, cloud computing controls were introduced into the framework in revision 4 [13]. To standardize the use of AI tools in cyber security for SMEs, they should be included in a cyber security framework. The NIST framework can be used as an inspiration for this [14]. Even with the implementation of AI, it is still necessary to use non-AI methods for security in parallel with AI, and paradigms, such as the zero-trust model (ZTM), are very useful for this [15].

The purpose of this study is to look at AI-powered security tools from an SME point of view, which is one of limited human and budgetary resources, where in the network these tools need to be located for maximum effect and to introduce them into a cyber security framework for SMEs.

The main methods centre around a literature review and a constructivist approach in developing the framework. This work starts by introducing the methodologies used and then by looking at small- and medium-sized enterprises from a cyber security point of view. After this, some AI tools that are available for SMEs are introduced. Then a framework is introduced that takes AI systems into account. Finally, a discussion and the main conclusions drawn from this work are given.

2 Methods

2.1 Research Methods

In this study new knowledge is being created from old and existing knowledge, and so constructivism presents itself as a suitable research paradigm. Even though it was originally a theory of knowledge for educational sciences [16], it can be used in the science and technology fields as well [17]. This research approach is qualitative being based on classifications of methods and not numerical results or data. The research design will be based around a literature review, and the research method will be the search for AI methods in cyber security from literature.

This work is an expert evaluation but from the point of view of a person responsible for cyber security in a SME. As a cyber security practitioner in an SME, the author is to some extent a participant in the research. This is an example of a complicated soft system in which the context and learning by the human operator are important parts of the system. The opposite of this would be a hard system which would include only the physical devices and systems, and measurements made on them, and the human only has the role of engineering it [18].

2.2 Usability Index

Usability is an important part of any cyber security tool and especially in situations where the user is a non-expert. Usability has been studied extensively and guidelines have been given to increase it. Taking the guidelines given by Nurse et al. [19] into account, a usability index for these cyber security tools was developed. This will be called the artificial intelligence security tool usability index (AISTUI), and it gives points for documentation, research, license, general usability and proliferation. These points are detailed in Table 1. Since the AISTUI is subjective, it should be only used as a general guideline, but a higher AISTUI score can still mean that the tool has a higher usability from an SME operator point of view using the index values provided. Operators can modify the index value ranges to suit their particular needs. Problems in the tool, such as a lack of documentation, can be highlighted using this method. It must be noted that the AISTUI score is not a judgment on the usefulness or quality of the tool in general, but only of the usability of the tool from the limited perspective of an SME.

A good documentation set is important for SME implementation because the personnel may be non-experts in the cyber security field. Without a good documentation set, it is harder for them to implement the tool and easier for them to make configuration errors, which can cause security problems [20]. The importance of a good documentation set has been known for a long time [21], and this is why AISTUI awards a higher score for good documentation.

The AI tool should be based on solid research that is published and available to the personnel if they so wish to study the tool further. Cyber security tools are based

Table 1 Details of the artificial intelligence security tool usability index (AISTUI)

Criteria	Description	Points
Documentation	How extensive is the documentation	0—No documentation
		…
		5—Extensive documentation that is easy to understand.
Research	Is there any peer-reviewed research on the tool	0—No research
		…
		1—Extensively researched
License	What license does the tool use	0—Non-free license
		…
		3—Open source. GPL or similar
General usability	How easy is the tool to use?	0—Almost impossible to use
		…
		5—Very easy to use
Proliferation	Describes how far the tool has spread within the cyber security field	0—no one uses it
		…
		3—Almost everyone uses it

on trust, and this can be increased by publicly providing information on the research behind the tool. This is why AISTUI awards a higher score for publicized research, but it is weighted less than the other points.

Operating any kind of software incurs some amount of cost that comes from licensing fees, hardware costs, support and maintenance costs and other fees. Even though open-source software can be used with a license that does not cost anything, licensing fees are only a part of the total cost of software adoption. A result is shown in the study by Sanchez et al. [22], who studied the adoption of open-source software. There is still a certain lack of AI solutions for cyber security, so many SMEs and even larger corporations may risk increased costs in the future as they are forced to implement new tools [23]. At some point the license fees and other operating costs incurred by implementing the tools may become too much of a financial burden on the SME, especially if a subscription model is used. Hardware costs can also depend on the solution selected. A good example of this is the Linux operating system, which works on PC hardware, compared to Unix, which generally requires dedicated, more expensive hardware. AISTUI awards a higher score for lower license fees and open-source tools that work on hardware readily available. No points are awarded if an expensive license is required, and a maximum of three points can be awarded for open-source solutions utilizing a general license and that work on existing easily procured hardware.

General usability is important for the same reasons that a good documentation set is important for SMEs. A tool that is difficult to use leads itself to misconfigurations

and security threats that arise from those. SMEs are usually not in a position to develop tools themselves or utilize tools that require extensive coding to set up or configure. This is why AISTUI awards a high score for good usability.

Proliferation of a tool is a good feature because it means that a large user base exists that can provide peer-level support, which is especially important for an open-source tool when a support subscription is not being used. This is why AISTUI awards a larger score for a tool that has a large user base. Also, if there are reported cases of successfully mitigating threats, then a higher score can be set here.

2.3 Scope

Even though this work is formally directed at SMEs, the target can be any instance with limited resources to allocate towards cyber security and that still needs to secure assets in cyber space. These could be sports club or congregations.

AI usually refers to models and algorithms, and it is one field of the information sciences. Machine learning is a subset of AI in which a system is taught to identify patterns and make independent conclusions. Deep learning builds on machine learning by including many more layers for processing and abstraction to solve problems. In this study, AI will be used as an umbrella term for all machine learning and deep learning methods.

The term ‘tools’ is used in this study in place of the term ‘methods’. The reason for this choice of terminology is that the term ‘method’ can be confused with a specific algorithm or means of implementing AI, and this is not what is being studied in this work. Also, the purpose of this study is not to advertise or endorse any specific tool for any specific purpose.

Only tools that utilize true machine learning will be included in this study. Therefore, tools that utilize some kind of simple algorithm, like calculating averages and comparing values to those, will not be included. The actual implementation method used in the machine learning part of the tool is not important for this study and will not be reported.

Many software services such as a cloud-based firewall (firewall-as-a-service, FWaaS) can utilize machine learning in many ways that are invisible to an SME user. An example of this is the utilization of machine learning in load-balance optimization [24]. These kinds of services have been included in this study where applicable.

3 Small- and Medium-Sized Enterprises

The purpose of this part is to define the concept of a small- and medium-sized enterprise (SME), show how it differs from a large enterprise and show what challenges it faces from a cyber security point of view.

3.1 Definitions

DeLone [25] defined an SE as having a sales revenue of $30 million and less than 300 employees in his work on the utilization of computers in enterprises already in 1988. This definition of SEs has remained relatively consistent over the years.

In Finland the Finnish statistical authority, Statistics Finland (Tilastokeskus in Finnish), has defined an SME as having fewer than 250 paid employees and a turnover of no more than 50 million or a balance sheet total, which is the sum of the total liabilities and the total equity, of not more than 40 million. The SME also needs to fulfil the criterion of independence, which states that large enterprises are not allowed to own more than 25% of the stock or equity of the SME. A small enterprise (SE) differs from a medium-sized enterprise in that it has fewer than 50 paid employees and a maximum annual turnover or balance sheet total of 10 million [1].

In 2020 the total number of enterprises in the Statistics Finland database was 368,949, of which 342,307 (92%) fulfilled the employee criterion of a small enterprise [1]. Therefore, the state of the cyber security of SEs is not an unimportant one because they make up a majority of all enterprises.

3.2 Cyber Security in SMEs

SMEs are not always at the forefront of adopting new technology. Research into how computers can help small businesses has been ongoing since the proliferation of computers [25]. In his study, Delone [25] found that the use of on-site computers can help produce positive effects for the business in cases where some businesses utilized computers and some did not. When SMEs are able to adopt and use information and communications technology (ICT), it is possible for them to grow faster than comparable enterprises that do not [26]. This result is taken for granted in the present era, where almost everything is carried out with a computer, and the SE environment has also moved past the question of whether the Internet adds any value to SEs. But still SMEs are not the place where the latest technology is being used.

One of the main differences between an SME and a large enterprise is in the number of resources available for ICT management, both in the amount of capital and the number of employees. When the number of resources available for ICT in general is smaller, then it necessarily follows that the resources available for allocation to cyber security is also smaller. In general, SEs and SMEs have less resources to fight cyber-attacks even though the effects and risks are the same as for larger entities [2]. The size of an enterprise, either large or small, is not a security control against any form of attack. An SE is just as likely to come under attack as a large enterprise, and in fact, in a 2014 study, it was found that 30% of cyber-attacks targeted SEs [3].

In a review on the cyber security situation of SMEs in Australia, Tam et al. [27] found that more research is still needed, but this research should be specifically targeted at SMEs, which are often lumped in studies with larger corporations. They note that most of the existing cyber security solutions are geared towards larger corporations and need the expertise to configure and maintain that the ICT staff there have. Larger corporations can also form alliances and share information easily. On the other hand, SMEs can benefit from their small size by being able to make changes quickly, and they too could form alliances with other SMEs [27].

Cyber security practices are incorporated into the business if cyber security has been made a part of the business routine, for example, in the form of a designated budget, and the technology is uncomplicated [28]. This last point is important for the proliferation of AI tools. If they have a high usability, then they will garner a large user base within the SME community.

Based on a survey of several small and large enterprises, Gutierrez [29] found no significant differences between SMEs and large enterprises when looking at communication, competency measurement, governance, partnership, architecture and skills. However, there was a positive correlation between how plans are applied in the organization and the size of the organization.

In their review on the cyber security of SMEs, Alahmari et al. [9] found that cybercrime is not considered a threat in SMEs because they feel that they are small and thus uninteresting for cyber criminals. Secondly, bad cyber security behaviour in SMEs has been found to account for many cyber security threats [9] which has been found in other studies [30].

It is therefore not a lack of knowledge of cyber security threats and good security practices that is the problem in achieving a good state of cyber security but a negative attitude and reduced situational awareness [30]. Cyber security behaviour can be improved with training, and the attitude towards cyber security threats can be improved with awareness programs. The role of decision makers is very large in an SME, and hence the cyber security awareness needs to extend to them and not only to the operational personnel. Alahmari et al. [9] found that recent research indicates that SMEs need more information on cyber security vulnerabilities, rather than knowledge on implementation of certain tools.

A study by Rawindaran et al. [31] reveals that SMEs mostly have the appropriate security packages implemented. The problem is more that the SME does not have the skills and experience in place to utilize them properly or to their full extent. Another result of this study was that risks and the impacts of successful attacks were not understood well enough leading to miscommunications within the enterprise.

3.3 Existing Cyber Security Framework

The NIST framework can be utilized by an SME for cyber security management. The core functions of the NIST framework are identify, protect, detect, respond and recover [11]. Chidukwani et al. [32] found that most of the research into SME

cyber security is only analysis using qualitative methods and the focus has been on the identify and protect functions of the NIST cyber security framework. This framework is a chain going from these parts and continuing to detection, response and recovery. Very little study on the other existing functions has been made even though SMEs should definitely have the ability to identify risks, define assets to protect and recover from attacks. Future research in SME cyber security should be more balanced between all the functions of the NIST framework [32].

3.4 Network Topology

As a frame of reference, a simplistic example of an SME network topology is shown in Fig. 1. The local area network (LAN) contains all the network devices needed for the network to function. These can be wireless access points, routers, switches or modems. Local resources are network printers or local storage and, in the case of a larger enterprise, intranet servers and domain controllers. External resources are then external storage, software services and potentially any cloud-based authentication services that the enterprise uses. In most cases e-mail is also processed off-location.

The network can also contain mobile devices, which potentially have a network connection that bypasses the main corporate firewall. Mobile devices can be connected to workstations by a wireless network connection, by a Bluetooth connection, via USB or via some other connection. These connections can then come with their own security issues which could become a concern [33].

A workstation can be physically located outside the business offices and on the public side of the corporate firewall but still connected to the local network via a virtual private network (VPN) connection. This kind of decentralized office structure is becoming more common. All of the locations shown in Fig. 1 are potential targets for attack and the network has many attack vectors. Even in this simple case, the situation, from a cyber security point of view, is much more complicated than if all the resources would be located behind the firewall. Attacks on any of these targets using, for example, some kind of malware can block the network, install additional spyware to obtain more information, gain access to personal information or disrupt services in the network making them unavailable [34].

The cyber security AI tools should be inserted into the system to disrupt these malware mechanisms. Potential locations for the tools are workstations, mobile devices and local resources, the corporate firewall, independently on the network and on the resources in the cloud (Fig. 1). Whether or not the SME has any control over the tool being used is not a concern for this study. The SME might not have any control over an AI tool on an e-mail server in the cloud but might have full control over an AI tool in the local network.

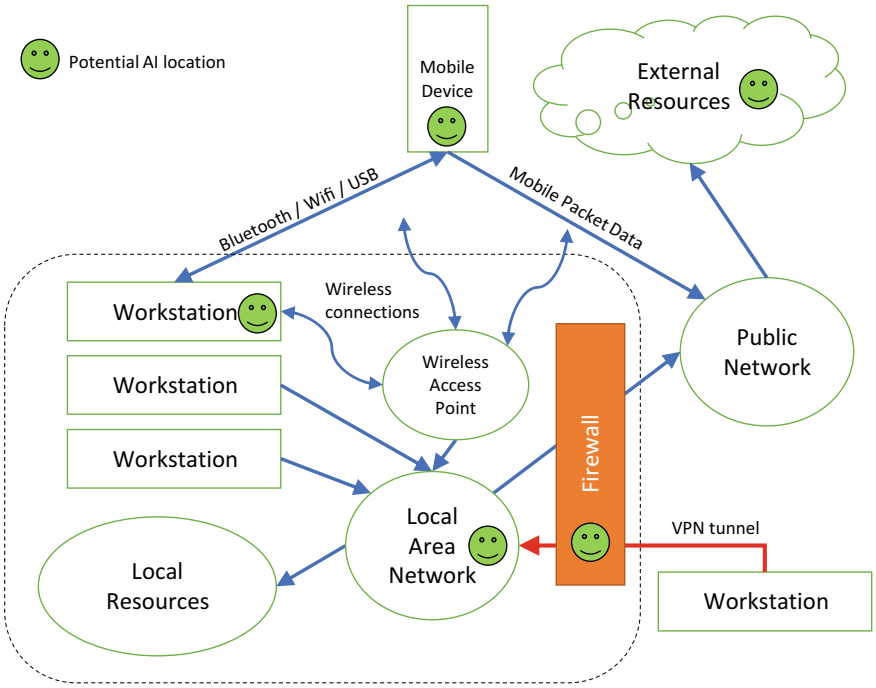


Fig. 1 Network topology of a typical small enterprise. Potential AI locations investigated in this study are shown with the green smiley face. The dashed line marks the physical boundaries of the SME office location. Note that a wireless access point can broadcast its signal outside this boundary

4 AI in an SME Environment

Of the core functions of the NIST framework [11], detection and response are most suited for an AI component because AI in general is good at finding patterns in large data streams. A well-functioning trustworthy AI tool could then be used as a first-response tool, but monitoring of the tool is the key to building trust in the tool [35]. Attack prevention is an almost impossible task, even for an AI, because of the lack of attack forecasting [36]. However, a well-connected AI system could pre-emptively react to an attack that is happening somewhere else. Protection is something that is accomplished by implementing the tool in the system and is something that the operator usually does.

The AI tools in an SME environment will be presented here following this location-based classification. The locations go from the local network to workstations and mobile devices, the firewall and then proceeding to external resources located on external servers.

4.1 Local Area Network

A few tools working in the local area network will be introduced first starting with a real-world example which is presented to show how the usability index AISTUI works and also to present some of the concepts related to using AI-based tools in an SME context.

The tool in this example analyses bandwidth through a network switch in three different but related channels. There is some legitimate system on the network that generates a lot of traffic at uniform intervals, which show up as spikes in the data. These spikes could be easily identified and filtered out using a statistical method, such as median filtering or implementing a threshold, but in this case, they are an important feature of the data rendering such statistical methods useless for anomaly detection.

Timeseries of data (blue lines in Fig. 2 show the underlying background patterns) were recorded, and the resulting data files were then fed to a deep-learning model developed with the Deep Learning Toolbox in Matlab. In the learning phase, the data was real data, and so it had all the normal small random variations associated with a real-world dataset but without any anomalies. After this learning phase, some anomalies were introduced into the timeseries in one channel to see whether the tool would be able to detect them. An anomaly next to a legitimate spike can be seen in Fig. 2. All the introduced anomalies were detected and marked correctly by the tool.

The usefulness of this tool for an SME from an operational point of view can be studied using the AISTUI index. The index generation starts with the score for

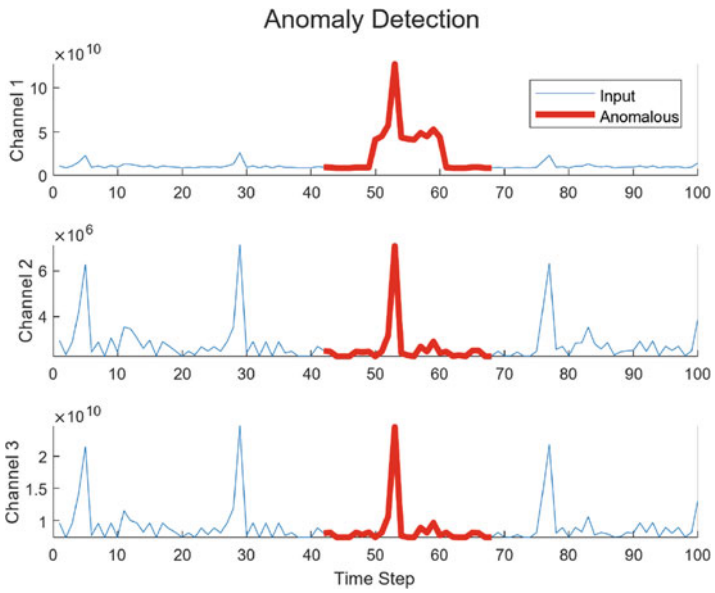


Fig. 2 Anomaly detected in timeseries. An example anomaly is in channel 1

Table 2 Usability index for a real-world example

Tool	Type	Document-ation	Research	License	General usability	Prolife-ration	Total score
Real-world example	IDS	1	1	0	1	0	3
Open-source version	IDS	1	1	3	2	0	8

documentation and for this only 1 point is given. This is not a judgement of the Matlab documentation, which is generally excellent and very extensive, but of the documentation of the tool itself. The tool has been developed from scratch, and the documentation for it is not generated automatically. Also, the data connections from the switch to the tool and back to the switch need to be developed, and the documentation for these needs to be generated. Because of this, only a score of 1 is given for documentation.

This tool is based on the deep learning toolbox which is based on extensive research into machine learning and deep learning. Therefore, a value of 1 is given for this point. Matlab licenses are far from free, and a deep learning toolbox license is required in addition to a Matlab license. For these reasons, a value of 0 is given for the license point taking into account the limited resources available to an SME.

This tool has a very steep learning and development curve, and so its general usability value in an operational setting is 1. This tool is better suited to a research context. Since this tool has been specially coded for this use, its proliferation score is 0. There is no user base or community for this tool unless it is published into the public domain and is taken up by other users. Again, this is not a judgement of Matlab which is widely used and has a broad user base.

The AISTUI values are summarized in Table 2 with the total score being only 3 out of a maximum of 17, which means that in a general sense this tool is not very usable in an SME setting. It must be noted that this score is valid only in an SME context and says nothing about the quality or functionality of the tool for other purposes.

This kind of tool could also be built around open-source packages. Python has a deep learning library that is relatively easy to set up and that is well documented [37]. Python scripts are also easy to integrate into a server environment for real-time use. If the above tool is reworked with Python, then the usability index score increases (Table 2). The usability problem still exists in the sense that everything needs to be coded in-house. Also, the documentation is still not generated automatically.

Online intrusion detection and offline security investigation have been applications of AI in the field of cyber security [38]. The real-world example above is a simple form of an anomaly-based intrusion detection system (IDS). In a general sense, an AI-based intrusion detection system (IDS) should be both signature-based and anomaly-based. If a machine learning algorithm is being trained in a compromised system, it will not detect the compromised network traffic as

anomalous. Therefore, a signature-based IDS or some kind of pre-taught system is required to give the system a chance to notice the breach during the training phase.

Shenfield et al. [39] developed an intelligent IDS system using the Matlab artificial neural network toolbox similar to the real-world example presented above. Their research was motivated by a real need to distinguish malicious shell code from benign code, which is something that signature-based IDS systems struggle with. Using a database of shell codes, they were able to teach their system to make the classification of code from offline files in an efficient way.

From an operational point of view in an SME, this tool would need to become an online tool that would be able to recognize malicious files from a data stream in real time. The system would then need to have the authority to make changes to the file system by deleting or quarantining the files detected as malicious.

The Palo Alto Cortex extended detection and response (XDR) system is an example of a commercial security solution for security operations centres (SOC) that utilizes AI, machine learning specifically, to distinguish new and emerging threats [8]. This system has successfully found the BlackCat ransomware, which is a sophisticated and constantly changing type of malware [40].

The Elastic Stack combines an open-source log mining tool, such as Filebeat, with the open-source Elasticsearch search engine and an open-source visualization tool, such as Kibana. This combination is referred to as the ELK stack. The logs that are mined can be generated by Wazuh, which is an open-source security information and event management (SIEM) system. Wazuh has been successfully used to detect security events in systems such as a web server [41].

Wazuh has the potential to produce a lot of information, some of which are false positives, and the AI methods available in the commercial version of Elasticsearch working in the cloud can be used to detect anomalies in that information. This is an example of an AI-powered offline IDS system used to detect possible attacks [42] but working off location on external servers (the cloud).

The ELK stack was integrated with a malware information-sharing platform (MISP) by Stoleriu et al. [43]. Using this system and the machine learning algorithms in Elasticsearch, they were able to perform real-time searches for indicators of compromise. This could be done looking at the network traffic in an online real-time way [43].

The usability index for IDS is shown in Table 3. The in-house developed intelligent IDS suffers from the same documentation and usability problems as the real-world example, so a few points have been deducted for those. Both examples of IDS tools require a commercial license, which is the Matlab license together with the machine learning toolbox license. Neither tool gets any points for that. A commercial tool developed by a large corporation will have a large user base and so is awarded full points for proliferation.

The offline cloud-based Elastic Stack system gets a few points deducted for the documentation, license and general usability criteria. This is because a subscription is needed and the tool is really only an offline system.

Table 3 Usability index for IDS

Tool	Type	Document-ation	Research	License	General usability	Prolife-ration	Total score
Commercial IDS	Network or workstations	5	1	0	4	3	13
In-house developed IDS	Network or workstations	2	1	0	1	0	4
Offline ELK stack	Cloud	3	1	1	1	1	7

Table 4 Usability index for operating system and application-level security tools

Tool	Type	Document-ation	Research	License	General usability	Prolife-ration	Total score
OS	Workstations	4	1	0	4	3	12
Applicationlevel	Workstations	4	1	1	4	3	13

4.2 Workstations and Mobile Devices

Microsoft Windows has a large market share in the workstation market with it being almost 75%in 2022 [44]. Windows security has so far been based on antivirus software, the Windows firewall, an Internet proxy service and Windows updates [45]. These are no longer sufficient by themselves, and additional tools are necessary to counter new and emerging threats. Security services based on machine learning and advanced AI are one new tool that could be used [45].

Microsoft Smart App Control blocks untrusted or unsigned applications. This control uses AI to predict whether an application will be safe or not and will block applications predicted to be unsafe [46]. As this is part of Windows 11, which is the recommended version of the most popular operating system, it is an example of AI working inside an SME.

Microsoft Defender for Endpoint has built-in threat intelligence. It utilizes the over 43 trillion security signals that Microsoft analyses every 24h to create a view of the threat landscape as it evolves over time. Artificial intelligence is used to analyse this data together with experts, and the emergent threats are then noted [46]. This is, however, an enterprise solution, and so it is perhaps not applicable to an SME.

The usability indii for these are shown in Table 4. Microsoft documentation is very extensive but can sometimes be hard to understand at a quick glance and so a point is deducted for that. These tools also require commercial licenses which mean they do not get any points for that category.

Signature-based antimalware (or antivirus) software is not capable of countering new and emerging malware technologies. This new malware is capable of changing the hashes of its code (mutating hashes) and can use novel and complicated obfuscation mechanisms to fool antimalware. They also use intelligent malware components that are able to react to changes in the environment [5]. Modern

malware is therefore very much able to evade detection making the 'detect and respond' model obsolete. Tools are already being sold online that are purported to be fully undetectable by signature-based, behaviour-based or heuristic antivirus software [5].

Next-generation malware requires a next-generation antimalware solution, and AI can help with this. An AI-based antivirus software that analyses the software executable itself was found to perform well, being able to distinguish between benign code and malware executables 98% of the time. The antivirus software was also fast with an average response time of less than 0.1s [47]. Antimalware usability on personal devices is usually good, but infection can still occur due to direct user interaction [48].

The usability index for antimalware tools is shown in Table 5. Antimalware tools are very similar regardless of whether they are located on the workstation or separately on the network. Therefore, its sufficient to calculate only one AISTUI value for them. They lose 2 points from the license criterion because they require a bought license or a subscription but usually they are quite affordable.

4.3 Firewall

Applications of AI have been found to work well in the field of online intrusion detection and offline security investigation [38], but they have been found to work in firewalls as well. A classical firewall is able to compare packets to predetermined lists of rules, while an AI-enabled firewall is able to make new rules when the situation warrants it [49]. Firewalls can be major sources of security threats due to poor configuration and poor usability [20].

A next-generation firewall (NGFW) has features beyond a normal stateful firewall that include cloud-based threat intelligence, application awareness and intrusion detection and prevention. An NGFW is capable of countering the many novel and complex types of attacks that target the network itself. Modern systems need to be aware of zero-day attacks and attacks taking over multiple steps and be able to counter them. Zero-day attacks utilize vulnerabilities that have not been made public. Attacks taking place over multiple steps occurs over several individual steps, some of which can be totally innocent. Statistical approaches, that can be rule-based or anomaly-based, can be used to find these attacks. Another approach is to use machine learning that learns to detect behavioural anomalies and the tracking of the sequence in which events occur [4].

Table 5 Usability index for antimalware

Tool	Type	Document-ation	Research	License	General usability	Prolife-ration	Total score
Antimalware	Network or workstations	5	1	1	4	3	14

Table 6 Usability index for firewalls

Tool	Type	Document-ation	Research	License	General usability	Prolife-ration	Total score
FWaaS	Cloud	4	1	0	2	3	10
NGFW	Network	4	1	1	2	3	11

Having physical firewall devices can be a strain on an organization and even limit the data throughput. By using network function virtualization in a firewall-as-a-service (FWaaS), the physical aspects of network perimeter security can be hidden from the user, for example, on a university campus [50] or an SME.

The usability indii for firewalls is shown in Table 6. Due to the nature of firewalls as devices that need to be fully configurable, they can be notoriously difficult to configure and prone to misconfigurations which can be easy to miss. This is true even in cases where the rules have been automatically generated by an AI-based system [51].

FWaaS-based systems always require a subscription to work, but some functionality is available in physical NGFW devices even without a subscription. In that case, it has limited use of AI-generated rules and signatures. Also, high hardware costs lower the score in this category.

4.4 Public Network Resources a.k.a 'The Cloud'

Modern SMEs have resources located on external servers (i.e. 'in the cloud'). Also, software-as-a-service (SaaS) is becoming more and more common, and in addition to productivity software, this software can take up some of the functions previously done by on-site security software or even hardware. An example of a cloud-based offline IDS solution that utilizes AI, the Wazuh-Elastic Stack combination, was already presented earlier.

Many security solutions deployed locally have a form of cloud-based information analysis that utilizes AI. There is a two-way link between the local system and the cloud-based system in which event information is sent to the cloud, where it is analysed. If new signatures and rules are developed, then these are fed back to the local system. Microsoft receives information on trillions of security events and its security products from around the globe and is able to use this to evolve its products [46].

The OpenStackDP system is a comprehensive security framework for cloud computing that is comprised of sensors monitoring network traffic and sensors specifically trying to detect anomalies. These work in the cloud close to the user data. An AI threat detection engine inside network devices looks for threats. Defensive actions are then taken that work using virtual network functions (VNFs) to make changes to traffic flow rules [52].

E-mail is a good example of a service located in the cloud because SMEs rarely have their own e-mail servers. E-mail servers need to be hardened and secured, and

Table 7 Usability index for AI in e-mail

Tool	Type	Document-ation	Research	License	General usability	Prolife-ration	Total score
AI in e-mail	Cloud	4	1	1	5	3	14

AI can definitely help with this, but it is not the role of the SME to harden and secure them if they are external. The security issues that e-mail brings with it are actually the same regardless of where the e-mail server is located. These issues are related to the content of the e-mail messages themselves, which can contain phishing emails, malware hidden in attachments, and unsolicited advertisements (a.k.a as spam e-mail) that can achieve large volumes.

E-mail classification into wanted and unwanted messages using AI machine learning methods is transitioning to a classification into the types of unwanted messages. This is to help in identifying the unwanted messages as the senders learn new techniques to beat existing identification methods [53].

AI tools work mainly at the server end to filter out as much of this unwanted e-mail as possible. AI tools are good at pattern matching and detecting anomalies, and so they have become good at blocking these kinds of unwanted e-mail. This is a good example of AI working to secure SMEs without input from users in the SME.

AI powered by user feedback is used by Gmail to catch unsolicited e-mail and helps in identifying patterns in large datasets. This is necessary because the spam environment is constantly changing, and the changes could be rapid. To determine what is classified as spam, Gmail uses AI-driven filters that look at the characteristics of the IP address, the domains and subdomain names, sender authentication, and feedback from users on spam that gets through. User feedback seems to be the most important criterion in determining the classification [23, 54].

A usability index can be generated for AI in e-mail even though in most cases the AI is quite invisible (Table 7). A few points have been deducted from a perfect score. Firstly, it is sometimes hard to find the work done by the AI from the documentation, and secondly all systems require a bought license, and most require a subscription.

It must be noted that having virtual servers in the cloud instead of physical servers does not make the cyber security situation very much different from having physical servers. Only the physical security part is outsourced, and many network security controls may be in place, but the server should still be secured like any physical server if it has services visible to the public Internet.

5 Framework

5.1 Zero-Trust Model

In a zero-trust model (ZTM), nothing is ever trusted without authentication and authorization. This helps in minimizing the damage if a breach occurs, because the

attacker will not have unlimited access to network resources by default. They will only be able to access those limited resources that the instance is authorized to access. In a ZTM, even the home network is considered to be hostile [15]. A ZTM model helps any AI working to secure the systems because they then do not have to make any assumptions about the network and can treat everything as suspect and subject to the same authentication and authorization rules.

In a classical cyber security view, the resources behind the corporate firewall connected to the local area network, the area inside the dashed line in Fig. 1, would be thought to be harmless and therefore trusted. Due to the many ways in which the local resources can be accessed, and the increasingly more complicated and developing threat environment, this is no longer the case. The ZTM should be incorporated into any framework that is used in an SME environment in addition to any AI tools in place.

5.2 AI in a Cyber Security Framework

The core functions of the NIST framework are identify, protect, detect, respond and recover [11], and these can be used as an inspiration for an AI-based framework [14]. AI could benefit SMEs in all of these functions, but mostly in the protect, detect and recover functions, and especially as a continuous cycle between the detect and respond functions (Fig. 3). The implementation of AI can itself be thought of as being a part of the protect function. A roadmap for AI implementation depending on attack type is given by Abdullahi et al. [4].

From the analysis shown previously, it is obvious that the non-AI NIST framework is no longer sufficient as it is and needs to be updated to take into account the next-generation malware. The following points could be considered to

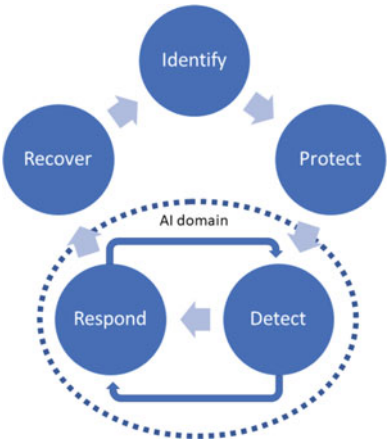


Fig. 3 The main functions of the NIST framework together with AI domain. The AI system would be most effective as a cycle between the detect and respond functions

be included in an AI-enabled version of the framework in the protect, detect and respond functions:

1. A NGFW or FWaaS that has machine learning capabilities needs to be in place and functioning as the corporate firewall.
2. An antimalware solution utilizing AI for malware detection should be used on the computer.
3. A ZTM should be implemented in all parts of the network.
4. The AI systems need to have authority to act on anomalies detected

This proposed framework is very much in line with the Palo Alto Networks approach to SOC operations that utilize AI in their Cortex XDR security product. This hinges around these three actions [8]:

1. Prevent all attacks that are possible to prevent
2. Rapidly detect attacks that cannot be prevented using AI methods or otherwise.
3. Automate future responses using AI-generated lists of rules.

The first point just means that all available AI and non-AI means of defence should be used. Even though attack methods are developing, it is still possible to get compromised by an older type of attack if the necessary security controls are not in place.

Attacks that then do get through need to be detected immediately and stopped. This requires the AI SOC to be authorized to make changes to the system, by deleting or quarantining files, by stopping processes and cleaning them from memory and by changing network rules. Since AI is based on a machine that is able to learn, then the successful attack can be used to teach the AI SOC to defend itself against the attack, and even attacks similar to it, in the future.

6 Discussion of Potential Problems

6.1 The Battle of Good AI vs Bad AI

There are challenges related to using AI for cyber security that SMEs need to be aware of before introducing them into their environment. Even as AI-based tools become available for SMEs, AI-based threats are also increasing. These two morally opposite manifestations of AI technology will start to confront each other more in the future.

If an AI system is introduced into an environment that already has malware working in it, it can potentially accept the malware as normal whilst it is learning the environment. Therefore, the new system needs to be pre-taught before being introduced into the environment or have some signatures that are potentially able to recognize the malware.

Malware can be introduced to hack the AI learning process. The internal objective function could be designed in such a way that it could have unintended consequences if subjected to data that is suspect [55]. This is called data poisoning, and it is a major concern for operators of machine learning and deep learning systems. In one of the worst-case scenarios, the AI system could grant a malicious actor full entry and exit rights to the system that it is guarding because it makes a wrong judgement call based on erroneous data [56].

AI models need a specific kind of defence that combats adversarial machine learning and preserves the integrity of the teaching data. After looking at the literature surrounding AI systems in cyber security, and AI-based attacks against them, Li [57] builds on existing research and shows how to build AI systems that are hardened against the threat of malicious AI.

Anderson et al. [58] review the different methods used to try to fool AI-based malware detectors. In their example, the malware was inside a Windows portable executable (PE) file. A series of attacks was made against the detectors to extract information about the AI model being used. Even a binary black box attack gives useful information about the ML system with binary in this sense meaning that the malware is either just detected or not detected. Even if the attacker has some knowledge of the system they are attacking, the number of times the attacker has been able to fool the malware detector has stayed low. The easiest way for the adversary to try to evade the malware detector would be if they were able to get the AI model being used.

A better way was then investigated by Anderson et al. [58] using a reinforcement learning (RL) agent that was able to change the PE file and present it to the malware detector. It was then able to iteratively test the malware detector trying to find blind spots in it in an automated way. A RL model consists of a learning agent working within an environment from which it is trying to extract information. The environment needs to have a measurable state that can be fed back to the agent. For each iteration, the agent chooses one action from a predefined set. The selected action can cause a change in the state of the environment, and this new state is used to calculate a reward for the agent. The reward and new state of the environment are given to the agent, which then determines what modifications the agent will make to the malware for the next iteration [58].

Adversarial examples can also help malware evade detection by interfering with the learning process of AI-based detectors. Adversarial examples are specialized inputs that confuse the artificial neural network, causing incorrect classifications. In malware this is accomplished by making changes to the computer code. It will still retain its malicious functionality but be classified as benign. Only a comprehensive analysis of the code of the adversarial example can show that the classification is wrong, and only then can the detection capability of the antimalware be improved [59].

6.2 Challenges Related to AI in Cyber Security

AI in cyber security for SMEs is not without its challenges and disadvantages. The AI information is available to attackers, and they use it to develop even more sophisticated malware that is able to circumvent the most modern AI-based defences.

The prevalence of tools utilizing AI for cyber security within SMEs is not as broad as it could be, even in developed countries. The challenges related to the adoption of such systems still revolve around human intervention and the need for specialist skills in implementation [31].

A problem in implementing AI tools is the lack of a suitable test environment in an SME environment. For example, using a simulated DDoS attack on a real system for stress testing purposes could render it totally inoperable.

AI implementation is not free and in some cases it can be expensive. Because the proliferation of AI solutions for cyber security is not so extensive, and there is a lack of easy-to-use cyber security tools, many businesses may have to spend more money on cyber security in the future. This includes SMEs. Also, as cyber threats evolve, cyber security tools will be updated leading to maintenance costs. AI systems cannot be left unattended, and, as with any ICT system, they need constant maintenance [23].

A big problem with AI that has been around as long as artificial neural networks have been around is the fact that AI systems are basically black boxes. The underlying decision-making capability is not in a human-understandable form. The explainable artificial intelligence (XAI) is an important new concept that has been formed with the intent of increasing trust in AI by allowing cyber security experts to study and understand the data evidence used and the causal reasoning that is implemented [60, 61].

In a sense, AI systems have a grasp of many concepts related to cyber security and thus very rarely exhibit a lack of understanding. However, they show a lack of judgement, meaning that they know about all the necessary concepts but still make wrong classifications. An example of this kind of artificial stupidity would be a face recognition algorithm that classifies a face on a shirt as a face [62]. This needs to be kept in mind when assessing what AI systems to implement and where to implement them.

When the AI systems start to become more common, they will start to attract more and more attacks. This means that the systems themselves and the data that they use to learn has to be kept secure. It is possible that totally new security controls will need to be developed and implemented when AI systems are in use [35].

Taddeo et al. [35] propose three actions that help in building trust in AI systems: in-house development, adversarial training and parallel monitoring. They argue that it is more important to develop an AI system in-house because in the event of a breach of a cloud system, the AI model and data will be compromised. Adversarial training places similar AIs against each other to test their functionality

and robustness. In parallel monitoring, the system is continually monitored against a similar system to notice anomalous behaviour.

As has been seen before, due to the lack of resources, these points will be difficult for SMEs to accomplish. Therefore, they have no option but to have an inherent trust in the AI systems if they want to implement them. On the other hand, due to the nature of new and emerging threats, they will probably have no choice but to implement all the possible security controls, AI and non-AI, that they have access to and can afford.

7 Conclusions

Some artificial intelligence (AI)-based tools were studied from the viewpoint of a small- and medium-sized enterprise (SME) user. The size of an SME does not secure it from cyber-attack, and it is subjected to the same regulations as larger enterprises.

One important finding was that AI is already implemented in many tools that are available to SME users, and therefore, it is actually already working to secure SMEs if they have those tools implemented.

The usability of the tools is an important factor governing the proliferation of the tools within the SME community. An AI usability index (AISTUI) was developed to measure the usability of the tools.

The AI-based tools were then placed into an AI-enabled extension of the NIST framework. The AI tools work naturally as a cycle between the 'detect' and 'respond' functions of the framework. For this to work, the AI tools need to have a mandate to make changes to the system, either by modifying files or network rules.

A discussion of AI tools also showed that they are not without challenges for SME implementation. They are black boxes that need to be inherently trusted, and SMEs do not have the infrastructure to safely test them in a proper environment.

Due to the rapidly changing threat environment, SMEs can be left vulnerable if they do not implement all controls, both AI and non-AI that they are able to.

References

1. Tilastokeskus: Small and medium size enterprises (2022). https://www.stat.fi/meta/kas/pienet_ja_keski_en.html. Cited 15 May 2022
2. Selznick, L.F., LaMacchia, C.: Cybersecurity liability: how technically savvy can we expect small business owners to be. J. Bus. Tech. L. **13**, 217 (2017)
3. Raghavan, K., Desai, M.S., Rajkumar, P.V.: Managing cybersecurity and ecommerce risks in small businesses. J. Manag. Sci. Bus. Intell. **2**(1), 9–15 (2017)
4. Abdullahi, M., Baashar, Y., Alhussian, H., Alwadain, A., Aziz, N., Capretz, L.F., Abdulkadir, S.J.: Detecting cybersecurity attacks in internet of things using artificial intelligence methods: a systematic literature review. Electronics, **11**(2), 198 (2022)

5. Scott, J.: Signature based malware detection is dead. Institute for Critical Infrastructure Technology (2017). https://informationsecurity.report/Resources/Whitepapers/920fbb41-8dc9-4053-bd01-72f961db24d9_ICIT-Analysis-Signature-Based-Malware-Detection-is-Dead.pdf. Cited 15 May 2023
6. Butavicius, M., Taib, R., Han, S.J.: Why people keep falling for phishing scams: the effects of time pressure and deception cues on the detection of phishing emails. Comput. Secur. **123**, 102937 (2022)
7. Li, B., Xue, C., Cheng, Y., Lim, E.T., Tan, C.W.: Understanding work experience in epidemic-induced telecommuting: the roles of misfit, reactance, and collaborative technologies. J. Bus. Res. **154**, 113330 (2023)
8. Paloalto Staff: Artificial Intelligence and Machine Learning in the Security Operation Center. Paloalto Networks, overview, Paloalto Networks (2020) Available via DIALOG. https://www.paloaltonetworks.com/resources/techbriefs/artificial-intelligence-and-machine-learning-in-the-security-operations-center. Cited 15 May 2023
9. Alahmari, A., Duncan, B.: Cybersecurity risk management in small and medium-sized enterprises: a systematic review of recent evidence. In: 2020 international conference on cyber situational awareness, data analytics and assessment (CyberSA), pp. 1–5. IEEE, Piscataway (2020)
10. Aggarwal, K., Mijwil, M.M., Al-Mistarehi, A.H., Alomari, S., Gök, M., Alaabdin, A.M.Z., Abdulrhman, S.H.: Has the future started? The current growth of artificial intelligence, machine learning, and deep learning. Iraqi J. Comp. Sci. Math. **3**(1), 115–123 (2022)
11. National Institute of Standards and Technology (NIST): Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1 (2018). https://doi.org/10.6028/NIST.CSWP.04162018
12. Roy, P.P.: A high-level comparison between the nist cyber security framework and the iso 27001 information security standard. In: 2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA), pp. 1–3. IEEE, Piscataway (2020)
13. Tariq, M.I., Tayyaba, S., Ashraf, M.W., Rasheed, H., Khan, F.: Analysis of NIST SP 800-53 rev. 3 controls effectiveness for cloud computing. Computing **3**(4) (2016)
14. Masombuka, M., Grobler, M., Watson, B.: Towards an artificial intelligence framework to actively defend cyberspace. In: European Conference on Cyber Warfare and Security, pp. 589-XIII. Academic Conferences International Limited (2018)
15. Kindervag, J.: No More Chewy Centers: The Zero-Trust Model of Information Security. Forrester Research (2016)
16. Piaget, J., Cook, M.: The origins of intelligence in children, vol. 8, No. 5, pp. 18–1952. International Universities Press, New York (1952)
17. Boudourides, M.: Constructivism, education, science, and technology. Can. J. Learn. Tech. **29**(3) (2003)
18. Checkland, P., Scholes, J.: Soft Systems Methodology in Action. John Wiley & Sons, London (1999)
19. Nurse, J.R., Creese, S., Goldsmith, M., Lamberts, K.: Guidelines for usable cybersecurity: past and present. In: 2011 Third International Workshop on Cyberspace Safety and Security (CSS), pp. 21–26. IEEE, Piscataway (2011)
20. Voronkov, A., Iwaya, L.H., Martucci, L.A., Lindskog, S.: Systematic literature review on usability of firewall configuration. ACM Computer Surv. **50**(6), 1–35 (2017)
21. Gemoets, L.A., Mahmood, M.A.: Effect of the quality of user documentation on user satisfaction with information systems. Inf. Manag. **18**(1), 47–54 (1990)
22. Sánchez, V.R., Ayuso, P.N., Galindo, J.A., Benavides, D.: Open source adoption factors-a systematic literature review. IEEE Access **8**, 94594–94609 (2020)
23. Dalave, C.V., Dalave, T.: A review on artificial intelligence in cyber security. In Proc. 6th Int. Conf. Comput. Sci. Eng. (UBMK), pp. 304–309 (2022)

24. Dezhabad, N., Sharifian, S.:Learning-based dynamic scalable load-balanced firewall as a service in network function-virtualized cloud computing environments. J. Supercomp. **74**, 3329–3358 (2018)
25. DeLone, W.H.: Determinants of success for computer usage in small business. In: Mis Quarterly, pp. 51–61 (1988)
26. Kamal, M.: Potential of cloud-based infrastructure for small business development. In: 2012 45th Hawaii International Conference on System Sciences, pp. 4860–4867. IEEE, Piscataway (2012)
27. Tam, T., Rao, A., Hall, J.: The good, the bad and the missing: a narrative review of cybersecurity implications for Australian small businesses. Comput. Secur. **109**, 102385 (2021)
28. Eilts, D.: An Empirical Assessment of Cybersecurity Readiness and Resilience in Small Businesses. College of Computing and Engineering, Nova Southeastern University (2020)
29. Gutierrez, A., Orozco, J., Serrano, A.: Factors affecting IT and business alignment: a comparative study in SMEs and large organisations. J. Enterp. Inf. Manag. **22**(1/2), 197–211 (2009)
30. Ncubukezi, T.: Human errors: a cybersecurity concern and the weakest link to small businesses. In: Proceedings of the 17th International Conference on Information Warfare and Security, p. 395 (2022)
31. Rawindaran, N., Jayal, A., Prakash, E.: Machine learning cybersecurity adoption in small and medium enterprises in developed countries. Computing **10**(11), 150 (2021)
32. Chidukwani, A., Zander, S., Koutsakis, P.: A survey on the cyber security of small-to-medium businesses: challenges, research focus and recommendations. IEEE Access **10**, 85701–85719 (2022)
33. Hassan, S.S., Bibon, S.D., Hossain, M.S., Atiquzzaman, M.: Security threats in Bluetooth technology. Comput. Secur. **74**, 308–322 (2018)
34. Ahsan, M., Nygard, K.E., Gomes, R., Chowdhury, M.M., Rifat, N., Connolly, J.F.: Cybersecurity threats and their mitigation approaches using machine learning—a review. J. Cybersecur. Priv. **2**(3), 527–555. (2022)
35. Taddeo, M., McCutcheon, T., Floridi, L.: Trusting artificial intelligence in cybersecurity is a double-edged sword. Nat. Mach. Intell. **1**(12), 557–560 (2019)
36. Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A.V., Di Franco, F.: The role of machine learning in cybersecurity. Digi. Threats: Res. Pract. **4**(1), 1–38 (2023)
37. Chollet, F.: Deep learning with Python. Simon and Schuster (2021)
38. Parrend, P., Navarro, J., Guigou, F., Deruyver, A., Collet, P.: Foundations and applications of artificial intelligence for zero-day and multi-step attack detection. EURASIP J. Inf. Secur. **2018**, 1–21 (2018)
39. Shenfield, A., Day, D., Ayesh, A.: Intelligent intrusion detection systems using artificial neural networks. Ict. Express **4**(2), 95–99 (2018)
40. Tanner, D.A., Hinchliffe, A., Santos, D.: Threat assessment: blackcat ransomware. Palo Alto Networks (2022). https://unit42.paloaltonetworks.com/blackcatransomware/. Cited on 15 May 2022
41. Stankovic, S., Gajin, S., Petrovic, R.: A Review of Wazuh tool capabilities for detecting attacks based on log analysis. In: Proceedings, IX International Conference IcETRAN, Novi Pazar, Serbia, 6–9. June 2022 (2022)
42. Negoita, O., Carabas, M.:Enhanced security using elasticsearch and machine learning. In: Intelli. Comput.: Proceedings of the 2020 Computing Conference, vol. 3, pp. 244–254. Springer, Berlin (2022)
43. Stoleriu, R., Puncioiu, A., Bica, I.: Cyber Attacks detection using open source ELK stack. In: 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1–6. IEEE, Piscataway (2021)
44. Statcounter: Desktop Operating System Market Share Worldwide (2023). https://gs.statcounter.com/os-market-share/desktop/worldwide. Cited 15 May 2023

45. Dunkerley, M., Tumbarello, M.: Mastering Windows Security and Hardening: Secure and Protect Your Windows Environment from Intruders, Malware Attacks, and Other Cyber Threats, 2d edn. Packt Publishing Ltd. (2022)
46. Microsoft: Windows 11 Security Book: Powerful security from chip to cloud, Microsoft, (2022). https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWMyFE. Cited 15 May 2023
47. de Lima, S.M., Silva, H.K.D.L., Luz, J.H.D.S., Lima, H.J.D.N., Silva, S.L.D.P., de Andrade, A.B., da Silva, A.M.: Artificial intelligence-based antivirus in order to detect malware preventively. Prog. Artif. Intell. **10**(1), 1–22 (2021)
48. Lalonde Lévesque, F., Davis, C.R., Fernandez, J.M., Chiasson, S., Somayaji, A.: Methodology for a field study of anti-malware software. In: Financial Cryptography and Data Security: FC 2012 Workshops, USEC and WECSR 2012, Kralendijk, Bonaire, March 2, 2012, Revised Selected Papers, vol. 16, pp. 80–85. Springer, Berlin (2012)
49. Chakraborty, P., Rahman, M.Z., Rahman, S.: Building new generation firewall including artificial intelligence. Int. J. Comput. App. **975**, 8887 (2019)
50. Häberle, M., Steinert, B., Menth, M.: Firewall-as-a-service for campus networks based on P4-SFC. In: Electronic Communications of the EASST, vol. 80 (2021)
51. Alicea, M., Alsmadi, I.: Misconfiguration in firewalls and network access controls: literature review. Future Internet **13**(11), 283 (2021)
52. Krishnan, P., Jain, K., Aldweesh, A., Prabu, P., Buyya, R.: OpenStackDP: a scalable network security framework for SDN-based OpenStack cloud infrastructure. J. Cloud Comput. **12**(1), 26 (2023)
53. Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E.: A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artif. Intelli. Rev. **56**(2), 1145–1173 (2023)
54. Kumaran, N.: Understanding Gmails SPAM filters, Google (2023). https://workspace.google.com/blog/identity-and-security/an-overview-of-gmails-spam-filters . Cited on 14 May 2023
55. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in AI safety (2016). arXiv preprint arXiv:1606.06565
56. Ahmed, I.M., Kashmoola, M.Y.: Threats on machine learning technique by data poisoning attack: a survey. In: Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25, 2021, Revised Selected Papers, vol. 3, pp. 586–600. Springer, Singapore (2021)
57. Li, Jh.: Cyber security meets artificial intelligence: a survey. Frontiers Inf. Technol. Electron. Eng. **1**, 1462–1474 (2018). https://doi.org/10.1631/FITEE.1800573
58. Anderson, H.S., Kharkar, A., Filar, B., Roth, P.: Evading Machine Learning Malware Detection. Black Hat (2017)
59. Li, X., Li, Q.: An IRL-based malware adversarial generation method to evade anti-malware engines. Comput. Secur. **104**, 102118 (2021)
60. Mahbooba, B., Timilsina, M., Sahal, R., Serrano, M.: Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. Complexity **2021**, 1–11 (2021)
61. Schmidt, P., Biessmann, F., Teubner, T.: Transparency and trust in artificial intelligence systems. J. Decis. Syst. **29**(4), 260–278 (2020)
62. Eilts, D.: An Empirical Assessment of Cybersecurity Readiness and Resilience in Small Businesses. College of Computing and Engineering, Nova Southeastern University (2020)

On the Cybersecurity of Logistics in the Age of Artificial Intelligence

Mikko Kiviharju

1 Introduction

Operational technology (OT) refers to industrial automation and control systems, such as SCADA systems, IoT, and the like. The official (draft) ISO definition [7] for OT is "technology for detecting, managing or causing change through the monitoring or control of a physical entity."

OT lies in the heart of many transport and logistics sector infrastructure, such as aviation, maritime, railways, and supply chain systems. Each of these logistics domains is under a constant stream of cyberattacks worldwide, as shown by recent OT threat reports [15] and related CI subdomain incident maps [6], and the rate is increasing.

Artificial intelligence and its most prominent implementation type, i.e., machine learning, have risen in the last decade into a whole new paradigm to build different ICT software components. This new paradigm is today present in traditional office IT systems as well as industrial automation systems. Many of these IT/OT systems are controlling infrastructure that is considered part of CI. It is then desirable to have a high level of assurance on the cybersecurity of these components. There is, however, a growing number of new unexplored inter-system interfaces accompanied with a general lack of knowledge, how the components under this new paradigm behave under adverse circumstances.

It has been argued [1] that "in essence, AI is software and therefore software security measures can be transposed to the AI domain" to some extent at least. However, there are other aspects to consider as well:

M. Kiviharju (✉)
Computer Science Department, Aalto University, Espoo, Finland
e-mail: mikko.kiviharju@aalto.fi

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_9

1. Information security philosophy: the basic enabler for software vulnerabilities in traditional IT systems lies in the poor understanding of what the exact software implementation functionality is, with all the unintentional side effects. Companies and organizations go to extraordinary lengths to gain more assurance that a given software component completes its intended tasks well, and only those. However, in ML systems, there is (at least before the advent of realistic explainable AI) even less understanding of the exact borders of the functionality of the component. It is even one of the main goals of AI systems to be able to extrapolate and keep learning from new data, which then undermines one of the basic information security tenets of being able to exactly divide the system into secure and insecure states.
2. Dependence on data: aside from (rather static and well-known) configuration data, conventional systems cybersecurity can be considered fairly independently of the data they process. On the other hand, the functionality of an ML component is deeply intertwined with the data it processes, and the security aspects need to incorporate the content of the input as well.
3. Ecosystem-wide concerns: ML components are not just used as replacements, but also as a service. The theory and practice of secure outsourced computing is markedly younger than the field of information security, let alone secure outsourced, mediated, or federated AI systems.
4. System structural details: many neural network-based ML components work in a parallel configuration with a very large number of parameters, and the formation of these parameters is typically a black box for the developer. Thus, for the outside viewer, each inference activates the ANN nodes in an unpredictable way (until observed). Even if the I/O behavior and intended functionality of a software component stay the same when it is replaced by an ML-enabled version, the cyberattack surface may be drastically different from the conventional software version. Another example is the common requirement of logging the actions of critical security components. Given their black box type operation, it is not even clear what this logging requirement should produce to provide sufficient traceability.

The current research on cybersecurity of machine learning stems from generalization of known attack types, such as evasion or model extraction. There is still a lack of widely accepted and generalized theoretical security framework in the research of ML cybersecurity, resulting, e.g., in the incomparability of research results [1].

There is a recognized gap between general-level cybersecurity guidance for AI and domain-specific recommendations to securely operate AI systems as part of other systems—it is indeed unknown, where the border between general and domain-specific guidance should lie [1].

The task of protecting OT systems is heavily linked to safety aspects, which result in different or differently prioritized goals from the conventional information security CIA triad (confidentiality, integrity, and availability). Building systems that

will try to optimize partly contradictory goals is very delicate, and unexpected features and behavior are likely to occur, which again increases the cyberattack surface of these systems.

2 Modeling Cybersecurity of OT and ML

2.1 Protection Goals

The basic information security goals are centered around confidentiality, integrity, and availability (the "CIA-triad"). Extensions include more refined concepts of integrity to the list, such as authenticity and non-repudiability in information assurance (IA by NIST [9]). Authenticity can also be seen as the integrity of origin of message sender information and non-repudiation as the integrity of "who did it" or information of action (e.g., of using a digital signature key). Extensions of the confidentiality concept exist as well, such as confidentiality of personal identifiable information (privacy).

Often, the information security goal statement also depicts their priority: confidentiality of, e.g., business plans is considered primary to the correctness of language therein. However, when the information security is used within the broader scope of cybersecurity, other options such as system (e.g., database) integrity may be paramount.

Operational technology is primarily safety-oriented, and the information security protection priorities may be reversed and new ones added: the ISO standard for industrial control systems defines the CIA priorities to be in the order AIC [74], and often a fourth element, control, is added first [12]. Control refers here to the ability to control processes so that process state changes are both safe and secure. This set of priorities is referred to as CAIC.

In logistics, supply chain management plays a pivotal role, and the protection goals are again somewhat different (in this order): authenticity, integrity, confidentiality, and exclusivity, or AICE [13]. In the AICE priorities, authenticity refers to non-counterfeit components and exclusivity to the control of the chain by authorized stakeholders only.

Adding new goals becomes a problem, if some of the goals are contradictory to each other or their priorities do not match. This may have unexpected consequences in machine learning [8].

Machine learning presents itself as a technology to be utilized within IT and OT. Thus, the ML protection goals themselves are inherited from the domain of use. However, as ML dependency on data is markedly heavier than in conventional systems, protection of data integrity contributes more to overall system protection goals than in "plain" IT.

The German main standardization organization DIN noted that for ML-based OT, it may not be possible to require worst-case analysis from all possible situations

(which is the current norm in safety-critical applications), since the universe of all the possible states for an ML component is so huge. Instead, even safety situation-dependent assumptions checked at runtime would be more efficient [120].

2.2 Threat Models in ML

The discipline of cybersecurity inside machine learning is rapidly emerging and taking form, e.g., including the AI incident database (AIID, [17]) and MITRE ML-specific practical cybersecurity model (ATLAS, [18]). Furthermore, training and operating an ML system is a complex task that leads to numerous unintended functionality (misclassifications). However, our scope in this paper is intentional attacks.

Most cyber threats against ML systems stem from conventional threats against systems or system software. Some new types emerge due to the importance of data, including evasion, poisoning, backdooring, and model and data privacy [128]. Evasion, poisoning, and backdooring attacks are data integrity attacks at different ML data life cycle phases: evasion during the inference phase and poisoning and backdooring during the training phase. Backdooring attacks are more targeted than poisoning attacks in that they only target a smaller set of the training data.

The official threat models for machine learning [127, 128] are becoming more formal and inherit the model partitions from cryptography: adversarial knowledge of the target is separated as one independent axis, and the goal of the adversary is concretely described in classes, based on the attack. This is depicted in Fig. 1 below. Depending on the attack type, adversarial knowledge and goals can be formalized to a rather detailed level [21]. Some authors consider a third dimension with the attack type tailored for OT [39].

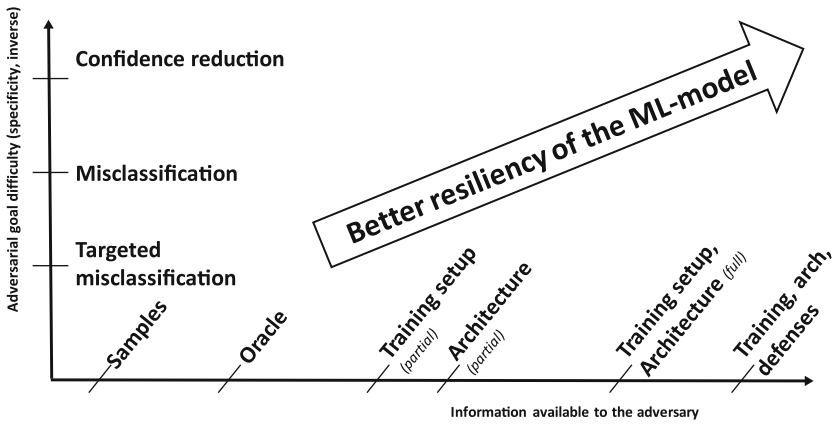


Fig. 1 Machine learning attack model for data integrity in evasion attacks. (Adapted from [128])

The adversarial knowledge available to the attacker can be categorized as follows:

- Black box: no information about the ML model is assumed, with only some samples of the ML decisions available or a stronger model with sufficient number of samples to estimate the output distribution (oracle access).
- Gray box: some information about the ML model and training setup is available.
- White box: all information about the model and its training is available, including the possible defenses.

Detailed attacker goals vary between attack types, but for, e.g., evasion and poisoning, the goals with increasing difficulty (or specificity and thus with larger impact) are as follows:

- Confidence reduction refers to any misclassification, failure to classify, or false decision by the ML model, whether it has deeper cybersecurity consequences or not or whether it happens with a high reliability or not.
- In misclassification, it is assumed the ML model will output an unintended but a valid classification/decision with a significantly higher probability than normal.
- Targeted misclassification allows the attacker to define to which class the output falls into.
- Evasion, poisoning, and backdooring attacks can each have all these attributes.

Adversarial attacks do not apply for all of the AI types but are mostly specialized for artificial neural networks (ANNs). For example, the Adversarial Robustness Toolbox implemented proof-of-concept adversarial attacks against AI [28], but it contains extra-ANN attack techniques (of the total of nearly 80 techniques) only for decision trees (3 techniques) and support vector machines (2 techniques). Completely missing are such categories as Bayesian approaches, K-nearest neighbor, and Q-learning. Other sources identify adversarial ML attacks also against reinforcement learning.

2.3 Attack Surface of Combined Technologies

The current mainstream practice of cybersecurity is based on conventional IT systems with their protection goals, standardized technologies, short life cycles, and specifically resourced and trained personnel. The OT systems work with the CAIC protection priorities instead of CIA and use sometimes very old legacy technology with no possibilities to upgrade. We argue that this creates a whole new attack surface domain that needs to be addressed independently from the IT domain.

Additionally, when we consider machine learning components, they are heavily dependent on data, especially in the learning phase but increasingly also in the operational (inference) phase. Due to this heavier interdependency, ML systems can also be attacked differently via manipulation of data. This creates conceptually

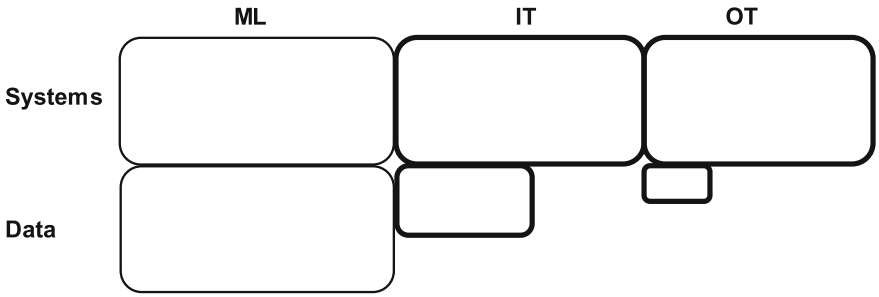


Fig. 2 Attack surface of different domains, with dependency of data

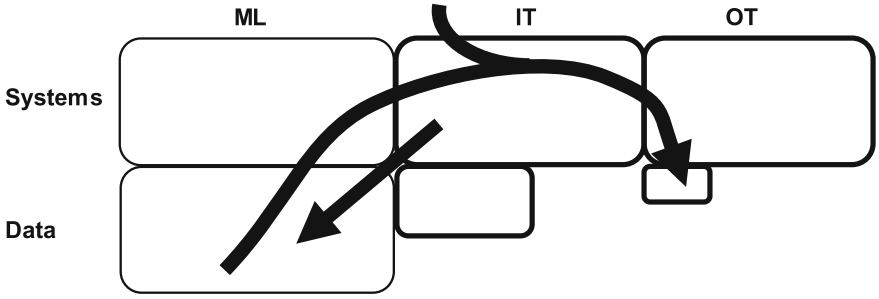


Fig. 3 An example of an attack path through different domains

yet another attack surface domain and the need to address data separately from the functionality of the system.

When these new domains are collected, the attack surface domains can be presented as in Fig. 2.

The attack surface for data decreases with the amount of data needed to make the system operational (configuration data), such as ANN node weights, a firewall rule list, or a preferred value range for an industrial control system actuator. Configuration data for OT is generally simpler than for IT systems, relative to system overall complexity.

These attack surface domains are not independent. Each domain contains internal threats, but there are also cross-domain threats: ML systems are used as intrusion detection systems (IDS) to look for suspicious activity and indicators of compromise (IOC). However, if these systems are themselves attacked, the implications are wider and may enable further attacker progress in the IT and OT domains. An example is depicted with SWaT (Secure Water Treatment—testbed OT system) [14], where a cyberattack against waterflow functions is camouflaged with an additional evasion attack against the IDS, the purpose being to fool the ML implementation inside the SWaT IDS to predict future behavior of sensors incorrectly, thus hiding the IOCs. The course of the attack through the attack surface domains is depicted in Fig. 3.

The ML implementations are mostly regular IT software and include conventional IT vulnerabilities, such as integer overflows, improper input validation, etc. (in [21], 21 different vulnerabilities concerning popular ML software libraries from 2018 forward listed in the CVE database were identified).

2.4 Emerging ML Kill Chains

The MITRE Corporation has developed frameworks to identify typical cybersecurity offensive and defensive operations tactics and techniques,[1] arranged by what is considered a typical cyberattack life cycle, "kill chain." The frameworks are constantly evolving since they are based on recognized attacks and research. The defensive framework is called D3FEND [20] and the offensive framework ATT&CK [19]. Neither framework catalogs procedure-level specifics of attack/defense methods.

MITRE has also gathered information on domain-specific attacks, such as for ML (called ATLAS [18]) and industrial control systems (part of ATT&CK). The ATLAS tactics are ML-specific techniques grouped according to ATT&CK tactics, with two ML-specific tactics additions: ML Model Access and ML Attack Staging (see Fig. 4). The ATLAS modeling is based on 16 case studies (e.g. [13]) as of the time of this writing. According to a survey [21], real-world cyberattack[2] stages against machine learning components roughly follow the life cycle suggested in ATLAS, so there is evidence that it realistically models the current state-of-the-art for the typical cyberattack against ML implementations.

From the data available in ATLAS, it is evident that currently both defensive and offensive strategies in cybersecurity consider the ML components as a black box: attack paths have an overall arch following conventional IT domain cyberattacks, but the ML-dependent attack tactics are addressed completely independently and their vulnerabilities and techniques only inside the ML domain.

A natural question is then whether the attacks against ML components in the logistics applications also fit in this model. However, as it will be explained later, there is too little data to answer this question even qualitatively.

[1] In the offensive model, tactics, techniques, and procedures (TTP) is the behavioral description of a cyberattack. Tactics describe the current and immediate generalized goal of the attacker, such as privilege escalation. Techniques, listed per tactic, are generalized known ways to achieve the tactic. Procedures are case-specific details of a particular technique.

[2] Taken from the ATLAS incident database and AIID.

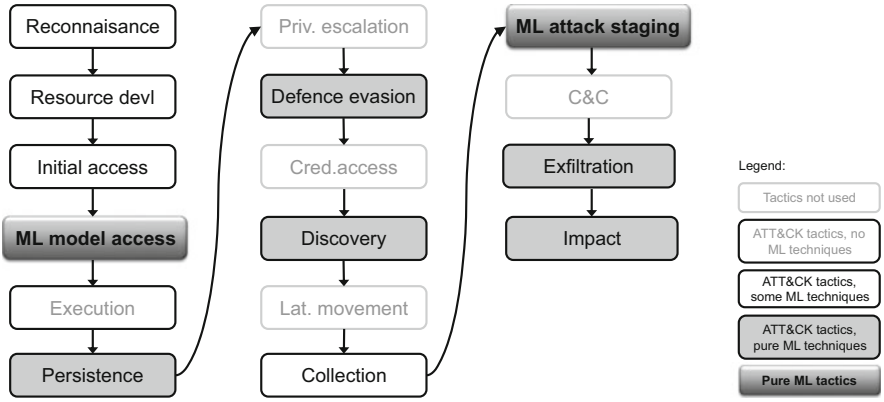


Fig. 4 Comparison of cyberattack tactics in MITRE ATLAS and ATT&CK frameworks

3 Cyber Threat Landscape in Logistics

3.1 General

In order to form a comparable risk profile of cyberattacks in logistics and/or transportation sector, information about known attacks is required. Statistics from different sources is listed below:

- Waterfall security has produced an OT threat report [15], which includes 116 different cyber-incidents against OT with physical consequences between 2010 and 2022. Thirty-five of these concern transport sector, and the overwhelming majority of them (27 cases) were ransomware attacks. In four cases, the threat actor was a nation-state or a politically motivated hacktivist group, and in five cases, the purpose or threat actor remained unknown.
- In an ICS incident database [22], 134 cyber-incidents[3] (not necessarily with significant consequences) between 1994 and 2023 were listed aimed at transportation, aviation, and automotive industries. Of these, 63 were ransomware attacks. Smaller portions included data breaches and website defacements. Many of the purposes or methods remained unknown due to limited disclosure.

There is then good evidence that for the recent years, ransomware forms the biggest threat to the logistics industry. This also indicates that the most common threat actors are criminals. Nation-states remain a significant threat, but with some exceptions they have not been interested in criminal cyber operations, although they may allow domestic criminal activity or even support them if the target nation is seen as adversarial.

[3] Plus 16 incidents listed as glitches.

From the point of view of tactics, however, the dominance of ransomware conveys too simplistic a picture: asking for a ransom is merely a criminal business model – the actual hacking tactics and techniques are as varied as with any other types of motive.

3.2 Domain-Specific Developments

To form a deeper picture of tactics and techniques used against logistics, domain-specific threat profiles present an alternative view.

ENISA monitors the cybersecurity of the transport sector on a regular basis. The report of 2023 covers 98 cyber-incidents in 2021–2022 [71], and the sector-specific number of incidents was as follows (with very little yearly variation):

- Aviation: 27%
- Road: 24%
- Railway: 21%
- Maritime: 18%

This indicates a rather evenly distributed threat and suggests that the need for protection is equally dire in all transport modalities.

Following the trend in [15], ransomware and financial gain were the main motivators, although attacks against maritime and railway sectors were performed for political purposes by nation-state actors as well. A peculiarity in the aviation sector was scamming using fraudulent airline websites.

The attack distribution between IT and OT systems was heavily centered in the IT systems, especially in the railway sector. However, incidents involving OT and ransomware were predicted to be on the rise. The data are in accordance with findings of domain-specific surveys [6, 23, 70]:

- In aviation, only two incidents (in [6]) revealed some connections with ML or OT: virus detection AI models used, e.g., by Air Italy being stolen in 2019 [67] and a data breach at US aerospace ICS technology subcontractor Parker Hannifin [68].[4]
- The most used points of attacks in maritime were ([23]) the privately owned operators' basic office IT infrastructure off-ship. Only about one-fifth of the total number of attacks involved OT systems as their attack points. ML components were not mentioned.

Cyberattacks against supply chains are usually discussed separately. Supply chain attacks aim to exploit more vulnerable organizations to get access to targeted, usually more secure organizations via the supply chain. According to ENISA report

[4] However, the published incident reports are very high level and rarely reveal individual technologies compromised.

analyzing 24 supply chain attacks from 2021 to 2022 [72], the attackers were mostly targeting suppliers' software (codebase) to attack the actual targets with malware or trust relationships to exfiltrate target confidential data, such as customer base and trade secrets/IPR. Supply chain attacks predominantly target conventional IT systems: the study contained no mention of OT, AI, or ML technologies.

3.3 Threat to ML Use Cases in Logistics

Machine learning in logistics and supply chain management serves mainly three purposes [25]:

1. Prediction (of demand, material flows, etc.)
2. Optimization (of material flows, fleet management)
3. Detection (of abnormal behavior, including cyberattacks)

Optimization of industrial processes may also require machine learning components during planning (IT systems) or during process monitoring (OT systems).

Additionally, there are many autonomous tasks that do not fit well under process optimization, such as autonomous vehicles (even though the autonomous system itself might be used for process optimization). We will thus adda a fourth category with autonomous vehicles/robots.

After identifying the main use cases, each of them is investigated, whether there are published attacks known to affect these systems particularly (excluding larger campaigns that have affected entire infrastructures) or if such have been described in related research papers. This is depicted in Table 1.

Smart port prediction and analysis with ML is a wide topic and includes such subtopics as (listed in [42]) demand prediction (e.g. container throughput), land and seaside operations (such as truck traffic and crane productivity), and safety (collision risk [41] and ship detention risk). The literature search included all of these subtopics.

Machine learning in supply chain management (SCM) can be divided into supplier selection and segmentation, risk management, demand and sales estimation, vehicle routing, demand prediction and inventory management [43]. Those areas of SCM, which are applicable for logistics scenarios, are addressed individually in Table 1. Sales estimation, supplier selection, and segmentation are deemed to be out of scope here.

The purpose of using machine learning in intelligent transport systems (ITS) is mostly the same across different modalities: situational awareness (location, surroundings, and possible objects), collision avoidance, and route/trajectory optimizations [58, 59, 62]. Modality-specific uses are as follows:

- Smart drones: for resource management, mobility management, surveillance and monitoring, power control and security management [58]

Table 1 Main use cases for machine learning in different logistics applications with domain-specific research and incidents

Use case	Theorized attacks	Known attacks
Prediction		
Arrival time forecasting [27]	(*None identified*)	(*None identified*)
Demand forecasting [26, 27]	[31, 32]	(*None identified*)
Traffic flow prediction [24, 27]	[34][a]	(*None identified*)
Location prediction [27]	(*None identified*)	(*None identified*)
User behavior prediction [26]	(*None identified*)	(*None identified*)
Predictive maintenance [24]	[35, 36, 38]	(*None identified*)
Smart port prediction and analytics [29]	(*None identified*)	(*None with ML identified*)[b]
Optimization		
Industrial process optimization (IT) [27]	[44]	(*None identified*)
Industrial process optimization (OT)	[69]	(*None identified*)
Vehicle routing [24, 27]	[45][c]	(*None with ML identified*)[d]
Intermodal transportation optimization [26]	(*None identified*)	(*None identified*)
Warehouse optimization [24]	(*None identified*)	(*None identified*)
Traffic management and policing [24]	[52]	(*None with ML identified*)
Traffic signal optimization [24]	(*None identified*)	(*None with ML identified*)[e]
Travel pattern recognition [24]	[53]	(*None identified*)
Detection		
Intrusion detection systems (e.g.[24, 29, 30, 37])	[14, 33]	[54, 55]
Transportation anomaly detection [27]	[57]	(*None identified*)
Anomaly detection in sensor data [24]	[14, 33, 37],	*Tools*[f] [56]
Autonomous vehicles		
Smart drones, drone swarms	[60, 61]	*ML tools and PoCs*
Self-driving (passenger) cars Package delivery robotic cars	[48, 51][g]	*Tools and PoCs*[h]
Driverless special-purpose cars (e.g., aircraft towing [24], road works protective vehicle [24])	(*None identified*)[i]	(*None identified*)
Autonomous ships	[62, 63]	*ML PoCs*
Autonomous trains [24]	(*None identified*)	(*None identified*)

[a]Adversarial activity on network traffic flow predictors is excluded here due to the different specifics of the data in both application areas

[b]There are documented cyberattacks against ports that can be deemed "smart" [40], but it is uncertain whether prediction and analytics components were affected

[c]The method is for a more general type of reinforcement learning, but the authors cite applicability to vehicle routing

[d]Vehicle routing/traffic management in general is a target for cyberattacks [47]

[e]There are cases, where the MQTT CAM messages to the traffic lights have been unauthenticated, but this is outside the scope of ML algorithms [46]

[f]Cleverhans used against IoT-IDS

[g]Surveys. Package delivery robots essentially share inference time threat with self-driving cars

[h]Even though many vulnerabilities have been demonstrated, an actual intentional attack is yet to be seen [50]. However, this is deemed as a real possibility [49], which is why the threat level is raised

[i]The adversarial resiliency of aircraft towing pathfinding ML types is researched in [65, 66], but not with airport data

- Drone swarms: for resource allocation (for the wireless channel and computing) and massive aerial access [59]
- Autonomous trains and ships: for decoding natural language information [62, 64]

Compiling Table 1, we have chosen to be very conservative on our criteria when an attack, attack tool, or research can be said to be applicable for a particular use case. We expect use case-specific simulations or data to be reported to qualify for a specific use case. Some care when reading the table is then advised: many of the researched attacks and tools may not require significant effort to transfer from one use case to another. However, ML implementations are typically highly dependent on training and inference data, which again is very use case dependent. We then postulate that unless there are use case-specific experiments or specific transfer learning methods used, there is still a sufficient barrier to cross for cyberattack techniques to affect some particular use case.

Additionally for Table 1, we have only considered attacks or research that aims to manipulate the system or component in some way. Thus, privacy-related attacks and data leaks or breaches are left outside the scope here.

Based on the survey by Akbari et al. [25], the main ML types used in ML for logistics and supply chain management are different types of artificial neural networks (ANNs), but a surprising number of more specialized or older technologies still prevail, such as Bayesian approaches, decision trees, fuzzy logic, and support vector machines (SVMs). As many of the modern cyberattacks against ML focus on ANNs and SVMs, some of the older or more specialized components are not under significant threat of advanced ML cyberattacks. The distribution of AI technologies is not particularly peaked on any specific logistics subdomain or purpose [26, 27]; instead age and individual problem scoping are more defining factors here. Bayesian approaches and decision trees are also very transparent in their operation ("explainable AI"), which contributes to the safety of the logistics process. A notable trend is that within the different logistics domains, ANNs are more popular in industrial process optimization than in other logistics domains.

From Table 1, systems that are only subsystems of a larger entity (such as arrival time forecasting for intelligent transport systems) are not primary targets for cyberattacks. However, critical subsystems (such as demand forecasting subsystems) may warrant research as part of risk analysis of the whole and demonstrate the attack potential for such systems.

Many of the applications where ML is used are interesting enough to warrant real-life cyberattacks, even though they do not (yet) target the ML components. The applications that are already targeted, but for which there is even no relevant research, are high-risk areas.

We categorized the use cases by the two attributes researched: whether there is use case-specific research on adversarial AI attacks or if there are additionally (or independently) known cyber-incidents on the use case in general or even some ML-specific tools and proofs-of-concept (PoC). This categorization is shown in Fig. 5, with the different use case types color-coded.

In the four-quadrant matrix of Fig. 5, we name the cells as follows:

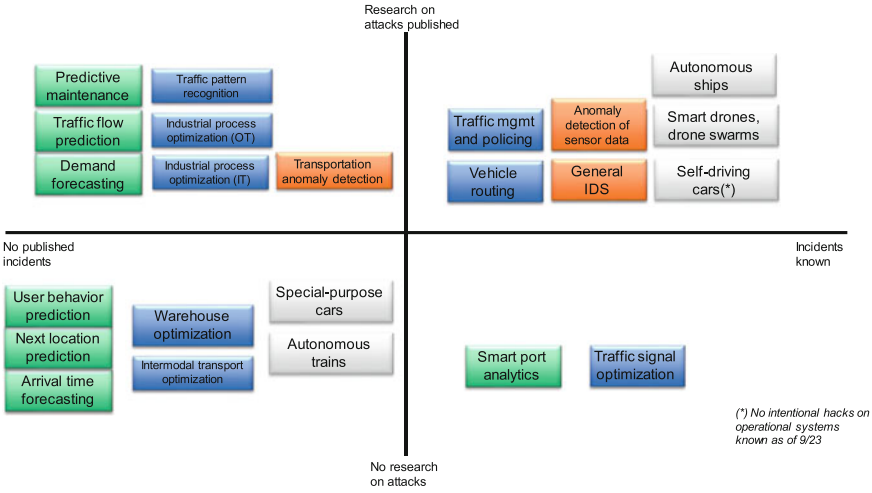


Fig. 5 ML use cases categorized according to the maturity of the threat

- Cell 1 (lower left), "adversarially uninteresting": application areas that are highly specific, very experimental to be in wide use, close to lower hanging fruits, or developed mostly as closed platforms. There is little research, or the research is too general for straightforward application. This cell contains mostly prediction and optimization applications, accompanied by some specific/closed autonomous systems.
- Cell 2 (upper left), "emergent": areas which are seen as interesting, but the adversarial techniques are still being developed. Tools development is still slow or constrained to targeted cyber operations under cover. This cell contains prediction and optimization areas that are more crucial to organizations and systems or closer to the public interfaces of the system.
- Cell 3 (upper right), "active": areas where the cyber arms race within the AI techniques is already happening. This cell does not include any predictive cases (yet), but some optimization and detection applications and many autonomous systems.
- Cell 4 (lower right), "blind spot": there are some use cases that are interesting to cybercriminals and cyber intelligence, but for which there is virtually no case-specific research or experiments. From a cybersecurity point of view, this cell should be sparsely populated, since attacks falling to these areas will face very little resistance.

When looking at the published cyber-incidents against machine learning components in logistics use cases, there are only relatively few of them. This may be due to mainly three reasons: (1) the whole area of practical adversarial machine learning is still emergent, and the majority of the existing hackers' toolset is still meant for non-AI purposes; (2) logistics and OT system life cycles are an order of

magnitude longer than IT systems, which means that ML components are not yet that common in logistics yet, and (3) logistics ML components are not usually very visible or immediately critical (prediction and optimization), which means that the motive for attacking them is mainly financial, and thus there needs to emerge a new and efficient way of monetizing the faults in ML in this context.

4 ML and OT in Normative Texts

4.1 General

Artificial intelligence and machine learning have been under intense normative work both in horizontal and sectorial standardization organizations [1] and also in governmental legislative and normative institutions. An EU project on following AI standardization [1] lists over 250 standardization initiatives already in 2021.

To understand how artificial intelligence and machine learning cybersecurity in logistics applications have been addressed in different standards and directives, it is instructive to have a glance from two sides of the picture: whether those normative texts intended specifically for AI/ML cybersecurity address any logistics applications and, on the other hand, if cybersecurity standards intended for logistics address any AI-/ML-specific issues.

Our assumption here is that general frameworks do not specifically address application domains (such as logistics), and we will thus focus on sector-specific standards. Our selection of general AI/ML cybersecurity standards will be narrower and based on the work in [1].

4.2 General Normative Frameworks on AI/ML

The general AI/ML framework standards selected for closer examination are based on the survey by ENISA [1]. The criteria used for selection is that the standard should address the technical cybersecurity of the ML components themselves on the data plane, i.e., discuss the security against ML-specific attacks, such as data poisoning or evasion. Many of the standards are under development, and in such a case, data may not be available, in which case we leave it out of scope in this survey.

In addition, we have selected some other prominent guidance for AI from NIST and national contexts (German DIN). Many of the texts do not yet have official standard's status but are either drafts or technical reports. However, they still offer an indication which AI/ML applications or application domains each standardization organization is considering. The findings are presented in Table 2.

In Table 2, the standardization organization name has been omitted in the standard name for brevity. From Table 2, AI cybersecurity standards do not yet

Table 2 Snapshot of AI/ML cybersecurity standards and their relation to logistics

Standard	Ref.	Description
International Standardization Organization (ISO)		
20547-4	[121]	ISO 20547-4 is the security and privacy portion of the ISO Big Data Architecture. This does not concern ML techniques per sé, even though ML components may benefit from the architecture
23894	[122]	ISO-23894 is a recent standard about AI risk management. It is dual to NIST AI RMF [118] and thus does not address application specific areas
TR 24028	[123]	The technical report 24028 surveys the concept of trustworthy AI more deeply but does not delve into application areas
TR 24029-1	[124]	Technical report 24029-1 researches the possible robustness metrics of ANN-based ML. It does not address logistics directly, but the specific metrics for image perturbations have relevance in many OT applications including self-driving cars
TR 27563	[125]	Technical report 27563 considers best practices for mostly privacy-related issues in ISO AI use cases [24], including logistics
AWI 27090	This standard is in predraft stage, and no information other than the title: "Cybersecurity – Artificial Intelligence – Guidance for addressing security threats and failures in artificial intelligence systems" is yet available	
TR 29119-11	[126]	Technical report 29119-11 is a part of software-testing documents, and this one focuses on AI software specifics to clarify the possible acceptance criteria for such systems. This is application area agnostic: it acknowledges the different use cases but does not elaborate on them. The next version of the document is still in predraft stage
European Telecommunications Standards Institute (ETSI)		
GR SAI-009	[116]	ETSI's Industry Specification Group (ISG) on Securing Artificial Intelligence (SAI) has published nine group reports (GR) as of the time of this writing, and GR no. 009 is the general framework. Like all the other documents in this ISG, this does not address logistics specific areas[a]
TR 103 674	[117]	ETSIs Technical Reports (TR) touch on various aspects not yet standardized. TR for SmartM2M architecture[b] no. 103674 discusses the effect of AI to one such specification. The TR addresses trustworthy and verifiable AI, on an architectural level, and suggests mitigation strategies outside AI techniques. Other SmartM2M documents do not concern AI/ML. No logistics applications are mentioned
National Institute of Standards and Technology (NIST)		
AI 100-1	[118]	The NIST Trustworthy and Responsible Artificial Intelligence Resource Center (AIRC) has been tasked the development of AI-related guidance in NIST. Currently it lists six documents under its website. The AI risk management framework (RMF, the AI-100-series) is a collection of three documents. The main document AI 100-1 is an application-agnostic guidance that instructs the users in managing AI-specific risks. OT or logistics are not specifically addressed

(continued)

Table 2 (continued)

Standard	Ref.	Description
AI 100-2	[8]	AI 100-2 is a companion document to AI 100-1, and it maps the ML-specific attacks and gives a taxonomy. It is also application agnostic, and logistics sectors are not addressed. AI 100-2 does, however, warn that using ML in the OT context is not as straightforward as in the IT context due to conflicting goals for the ML component
AI 100-3	[119]	AI 100-3 is an AI glossary. We list it here for completeness
Deutsches Institut für Normung (DIN)		
AI standardization roadmap	[120]	The German main standardization body (DIN) has produced a roadmap mapping the needs for AI standardization. It has special sections on industrial automation, mobility, and logistics. The roadmap calls for revision of OT-safety standards (especially IEC 61508) and ML testing and verification methods to be developed and standardized. Additionally, due to the safety-oriented security goals, and worst-case modeling complexity, dynamic risk assessment methods are recommended The roadmap does not consider cybersecurity-related issues for logistics applications but raises the need for interoperability for especially intermodal transport chains, which naturally will birth new seams to the total system creating new vulnerabilities

[a]The SAI ISG discusses "data supply chains," but this is synonymous to AI "data life cycle" rather than describing a logistics application sector
[b]SmartM2M, or machine-to-machine, essentially means OT

give ready solutions for logistics and OT. Some problem and research areas are recognized, and many general guidelines are likely applicable to logistics sectors, such as image perturbation metrics in ISO TR 24029-1 [124] for self-driving cars or ETSI TR 103 674 architectural advice on developing trustworthiness to AI application in the OT domain [117].

In a regulatory context, EU has recently stepped up its regulation of machine safety and security by updating the machinery directive 2006/42 to 2023/1230 [10], which will become binding in the beginning of 2027. In an EU study in 2020 [11], AI was recognized as a technology, which also affects machine safety significantly. This concern is addressed in the new directive by subjecting systems with "fully or partially self-evolving behavior using machine learning" to official third-party evaluation (notified bodies). This essentially means that to receive a "CE" marking, those products that use unsupervised learning even during inference need a specific approval.

The exact evaluation criteria for these components wait, as of this writing, the establishment of new EU AI regulatory framework. The latest proposal, however,

will place those OT components implied by the machinery directive in the "high-risk AI systems" category[5] [4].

The new machinery directive also addresses cybersecurity as a part of the safety requirements indirectly: control systems should be able to "withstand ... malicious attempts from third parties," and possible cybersecurity certificates can vouch for this property of systems.

4.3 Sector-Specific Standardization of OTCS

There are quite many operational technology standards that relate to cybersecurity (OTCS standards). These are briefly introduced in Table 3. The standards can be divided into horizontal and sector-specific standards as follows:

- *Horizontal standards*: horizontal OTCS standards do not address sector-specific issues and expect that standards within different sectors be based on the horizontal standard and not contradict it.
- *Railway systems*: a review study of OTCS standards in rail systems [2] found that the standards landscape is lagging behind other industries such as aviation. The standards in Table 2 concerning this sector are listed based on [2].
- *Maritime (including port) systems*: we included both onboard and portside standards and recommendations to our survey, listed in Table 3.
- *Aviation systems*: different aviation systems, including onboard, air navigation services, and air traffic management systems, are traditionally very carefully designed and tested in the safety aspects of systems due to the high impact of threats that actualize. The International Air Transport Association (IATA) has compiled a list of relevant aviation cybersecurity recommendations [5], which is used as a base document in Table 3.
- *Automotive systems*: the automotive industry has some existing standards for securing car-internal communications. However, when it comes to the totality of cybersecurity, more recent and consolidated standards emerge.
- *Supply chain security*: Risk management in supply chains (SCRM) has traditionally focused on organizational maturity for different types of risks from external organizations affecting the product or business. Direct technological risks center around detecting and avoiding counterfeit (and consequently low-quality) products, such as electronics. In a survey commissioned by US DOE [101], the author states that current cybersecurity SCRM standards and guidance texts do not adequately address the cyber threat for ICS in critical systems from the viewpoint of supply chain management. The survey lists 37 different international and US domestic standards and guidelines for cyber SCRM. Seven of these are related to OT or logistics and presented in Table 3.

[5] The levels are "low," "high," and "unacceptable," but the "unacceptable" systems are deemed to fall under criminal jurisdiction rather than formal evaluation

Table 3 List of most prominent OTCS cybersecurity-related standards and their relation to AI/ML

Standard	Ref.	Description
Horizontal		
NIST-SP800-82r3	[30]	*NIST Guide to Industrial Control Systems (ICS) Security* (Rel 3) was published in September 2023, and it is a policy-/architectural-level guidance on building a secure OT system. Major part of the standard is based on guidance on applying the NIST Cybersecurity Framework (CSF [108]) for OT. Although the guidance suggests using ML for the detection of cyber threats, it does not address specifically the cybersecurity of AI/ML components
ISA/IEC 62443	[74]	IEC 62443 is an important family of standards for OTCS. It is a binding reference at least in railway cybersecurity [2] and maritime [3]. The standards were established in 2021 as a horizontal standard. It is very comprehensive and addresses all abstraction levels from policies down to components, in 14 different documents (not all of them with a "standard" status). Although there are some portions in the standard that could be interpreted to be technology-specific (such as mobile devices and mobile code), AI and ML are not among them
Railway systems		
CENELEC TS50701	[76]	Standard by an EU standardization organization about railway cybersecurity risk management, based heavily on ISA 62443, with some technical elements as well. AI/ML not referenced
AS7770	[73]	This Australian standard is for cyber management and requirements for a secure system. AI/ML not referenced
DIN VDE V 0831-104	[75]	German standard for electronic signaling systems for railways, partly based on ISA 62443. AI/ML not referenced
NCSA-FI 517957	[127]	Finnish national security authority (Traficom) recommendation on cybersecurity in railways. Relies on the OT horizontal standards, NIST CSF, TS50701, and EU NIS directives. AI/ML not referenced
UIC guidelines	[78]	International Union of Railways guidance on information security management and best practices of train signaling and telecommunications within railways. References ISO 27001 and ISA 62443. Does not address AI/ML
ENISA good practices	[79]	An ENISA study that regards the level of implementation of cybersecurity measures in the railway sector, within the context of the enforcement of the NIS Directive (v.2016). Contains typical scenarios for railway cyberattacks. Does not reference AI/ML
CYRail D7.5	[77]	CYRail (CYbersecurity in the RAILway sector) was an EU Horizon 2020 project, whose recommendations on cybersecurity for signaling and communications onboard the train were considered influential in [2]. No references to AI/ML

(continued)

Table 3 (continued)

Standard	Ref.	Description
Maritime systems		
IMO MSC-FAL.1/Circ.3	[82]	International Maritime Organization's high-level recommendations on maritime cyber risk management to safeguard shipping, including functional elements supporting cyber risk management. A brief statement, referencing NIST CSF and ISO 27001. AI/ML not referenced
IACS R166	[83]	IACS has a consultative status within IMO responsible for the technical safety standards. Recommendation no. 166 on cyber resilience addresses technical design, construction, and testing aspects of ship's onboard OT systems. Does not reference AI/ML
IACS UR E26	[84]	IACS unified requirements are binding policies for ships constructed after each member country organization ratification. E26 provides a minimum set of requirements for cyber resilience of ships (as a whole). AI/ML is not referenced as of April 2022 version, which was withdrawn September 2023 to wait for next version due at the end of 2023
IACS UR E27	[85]	As E26, but for onboard systems and equipment. Does not reference AI/ML either
DNVGL-RP-0496	[86]	DNV (GL) is an accredited verification body. The maritime sector of DNV is a member body of IACS. This guideline is for cybersecurity management down to the technical level leveraging BIMCO and IMO guidelines. AI/ML not referenced
IEC 63154	[80]	Standard for maritime navigation and radio communication cybersecurity. Specifies requirements and methods of testing against cyber-incidents on a technical level, but does not reference AI/ML
BIMCO guidelines	[81]	BIMCO is an international shipowners' association. It contains guidelines on cybersecurity onboard ships based on high-level principles. AI/ML not referenced
ENISA port security	[29]	ENISA's document on best practices of port security and security measures from policy level to technical. Document recommends using machine learning to discover cyber indicators of compromise (IOC) in the cyber-SOCs, but does not address specifically the security of ML components themselves
Aviation systems		
EASA ED Dec.2020/006/R	[87]	EU Aviation Safety Agency is the main regulatory aviation body in EU. EASA's decision no. 2020/006 issued amendments to certain acceptable means of compliance (AMC) and/or guidance material (GM), to introduce cybersecurity guidance into certain certification specifications (CSs): aircraft cybersecurity was changed to include the requirement for protection against IUEIs,[a] and the Amendment 10 to AMC/GM Part 21 Issue 2 [88] talks about OT specific risks but does not deal with AI/ML

(continued)

Table 3 (continued)

Standard	Ref.	Description
EUROCAE ED202A	[89]	Many EUROCAE documents are used by EASA as airworthiness technical specifications. EUROCAE Documents (ED) 201-205 concern cybersecurity-related issues. ED202a concerns secure software development life cycle (SDLC) in cybersecurity in general. AI/ML is not addressed
EUROCAE ED203A	[90]	This EUROCAE document is about secure SDLC tool qualification, which does not include AI/ML
EUROCAE ED204A	[91]	The 204A is about secure operation of software products, and does not include AI/ML
EUROCAE ED205A	[92]	The 205A is about ground systems security, and does not include AI/ML
EUROCAE ED324	[93]	A new standard draft under development from EUROCAE WG114 for AI. This is a process standard for development and certification approval of aeronautical products implementing AI. Expected to finish by the end of 2024
ARINC 664	[94]	Aeronautical Radio, Inc. (standards development now under SAE International) makes industry standards for avionics equipment. ARINC 664 specifies an Ethernet variant for Aircraft Data Network. Different parts of 664 are specified from 2005 to 2019. No references to AI/ML
ARINC 823	[95]	ARINC 823 is about end-to-end ISO OSI layer 2 encryption and key management for ACARS datalinks (divided in two parts). Does not contain AI/ML considerations
ARINC 834-8	[96]	ARINC 834 defines a set of protocols and services to transmit avionics data across aircraft networks. Supplement 8 includes, e.g., data security enhancements. AI/ML not addressed
ARINC 835-1	[97]	The 835-1 document specifies digital signatures for downloadable software integrity, and does not consider AI/ML
ARINC 852	[98]	The 852 document is a security logging implementation guide based on NIST 800-92 and RFC 5424. AI/ML not referenced
ARINC 858	[99]	The 858 document specifies the Internet Protocol (IP) variant for aviation safety services and for message exchange to ground systems (divided in two parts). AI/ML outside the scope here
A4A Spec 42	[100]	Airlines for America make industry standards for aviation. Specification no. 42 is a technical and wide cybersecurity standard, encryption-heavy. AI/ML is not addressed here
Automotive		
SAE J3061	[102]	SAE International creates international technical industry standards mainly for automotive, but through acquisitions also for other sectors. J3061:2021 is recommended best practices for vehicle cybersecurity. This is a wide and relatively technical standard, but there is no notion of AI/ML

(continued)

Table 3 (continued)

Standard	Ref.	Description
ISO/SAE 21434	[103]	SAE 21434 is a cybersecurity management-level document, and does not address specific technologies, such as machine learning
ISO 21177	[104]	ISO 21177 specifies a security protocol between vehicle network elements, leaving AI/ML out of scope
UNECE WP.29 R155	[105]	United Nations' Economic Commission for Europe (UNECE) WP.29 produces vehicle regulations, which become binding for the contracting nations when they ratify them. Recommendation no. 155 is basically a type of approval regulation to use ISO/SAE 21434 for vehicle cybersecurity management system (CSMS). AI/ML is out of scope for this document
NHTSA Best practices	[106]	National Highway Traffic Safety Administration (NHTSA) has issued cybersecurity best practices for cars. This is a requirements document, very brief on technologies. AI/ML is, however, addressed briefly in clause G.6 to prepare for sensor risks also from the ML perspective
AUTOSAR	[107]	AUTOSAR (AUTomotive Open System ARchitecture) is a global development partnership of automotive industry, which publishes standards for software standardization, reuse, and interoperability for automotive electronic control units. It is basically a secure C++ middleware used also for cybersecurity. AI/ML is outside the scope, but AUTOSAR middleware can be used to implement ML functionalities
Supply chain security		
ISO 20243	[110]	The ISO 20243 document gives recommendations to the integrity of hardware and software ICT products, addressing the software supply chain. AI/ML is not of concern here
ISO-28004-2	[109]	The 28004-2 document is basically general security guidance for SMEs (small and medium-sized enterprises) to secure their supply chain
SAE AS5553D	[111]	The AS5553D document is a aviation-specific standard for electronic parts and supplier management, procurement, traceability, and control. AI/ML is not addressed
SAE AS6081A	[112]	AS6081A is similar to AS5553D, but it concerns distributors that purchase from the open market. AI/ML is not addressed here either
SAE ARP6178A	[113]	SAE ARP6178A describes a risk assessment tool for aviation counterfeit electronic parts. AI/ML is not part of the risk assessment issues
SAE ARP9134A	[114]	ARP9134(A) is a generic guideline for supply chain risk management for aviation; AI/ML is not mentioned
UL 2900-2-2	[115]	UL (Underwriters Laboratories) Solutions LLC is a US-based company specializing in safety (industry) standardization and certification. The 2900 standards series present general software cybersecurity requirements for safety-critical systems. Part 2-2 gives particular requirements for ICS. No AI/ML is addressed [88]

[a]Decision's term for "cyber": *IUEI* Intentional Unauthorized Electronic Interactions

Table 4 Overview of the OTCS standards in logistics (latest part publication year and relation to AI shown)

Horiz. standards	yr	AI
NIST-SP800-82r3	2023	AD
ISA/IEC 62443	2021	-
Aviation		
EASA ED Dec.2020/006/R	2020	-
EUROCAE ED202A	2014	-
EUROCAE ED203A	2018	-
EUROCAE ED204A	2020	-
EUROCAE ED205A	2022	-
EUROCAE ED324	2024	~F
ARINC 664	2019	-
ARINC 823	2007	-
ARINC 834-8	2020	-
ARINC 835-1	2014	-
ARINC 852	2017	-
ARINC 858	2021	-
A4A Spec 42	2020	-

Railways	yr	AI
TS50701	2023	-
AS7770	2018	-
DIN VDE V 0831-104	2015	-
NCSA-FI 517957	2021	-
UIC guidelines	2018	-
ENISA good pract.	2020	-
CYRail D7.5	2018	-
Automotive		
SAE J3061	2021	-
SAE 21434	2021	-
ISO 21177	2023	-
UNECE WP.29 R155	2021	-
NHTSA Best practices	2022	AD
AUTOSAR	2022	T

Maritime	yr	AI
IMO MSC.1/Circ.1526	2017	-
IACS R166	2022	-
IACS UR E26	2022	-
IACS UR E27	2022	-
IEC 63154	2021	-
BIMCO guidelines	2018	-
ENISA Port security	2019	AD
DNVGL-RP-0496	2016	-
Supply chain		
ISO-20243	2018	-
ISO-28004-2	2014	-
SAE AS5553D	2022	-
SAE AS6081	2023	-
SAE ARP6178	2023	-
SAE ARP9134A	2014	-
UL 2900-2-2	2016	-

Legend	F = Full standard	AD = Anomaly Detection
	~F = Full draft standard	T = tooling

Since "cybersecurity" as a concept is quite wide, many legal and user-centric documents were pruned from the list, trying to keep the focus on technical or technology-policy standards.

4.4 Overview

We have collected the overview of the data in Table 3 to Table 4. The sector-specific statistics of the OTCS standards are as follows:

- Horizontal standards: two pcs, between 2021 and 2023 (average 2022)
- Railways: seven pcs, between 2015 and 2023, average being 2019
- Maritime: eight pcs, between 2016 and 2022, average 2020
- Aviation: 13 pcs, between 2007 and 2024, average 2018
- Automotive: six pcs, between 2021 and 2023, average 2022
- Supply chain, between 2014 and 2023, average 2019

The number of standards is only indicative, since this is not a complete list. The average year of the standards is more interesting: the overall average is from 2019, making them about 4 years old. The largest variation can be seen in the aviation sector, but this may be due to the fact that airworthiness requirements have required more stringent and formal information security technological standards

early on. This variation and the number of investigated standards also make aviation standards the oldest within different sectors. Somewhat surprisingly, automotive OTCS standards are the most current.

The OTCS standards do not address machine learning at all or by merely recommending using ML-based IDS. Some of the reasons are obvious, such as the scope of the standard: ISO 27001, ISA/IEC 62443, and CENELEC TS 50701 are agnostic to many of the implementation level issues and will likely never need to address AI/ML technology-specific cyber threats. Adversarial machine learning has not been a significant issue in operational implementations mainstream before 2020, which is why older standards are not likely to acknowledge the technology in any case. Furthermore, AI/ML in logistics OT still relies on AI technologies outside ANNs, which may make many of the new attacks moot. From Tables 3 and 4, it is also to draw the conclusion that standardization organizations have so far been busier with OTCS standardization in general, not OTCS-ML issues specifically.

Having surveyed the logistics, OT and AI/ML standards, and guidance from both general frameworks and sector-specific sides, it can be concluded that the intersection of machine learning, OT, and cybersecurity is still very much under research, and not in standards in any way. There are some developments in any case, and the situation may evolve rapidly by 2025, judging by the expected publication dates of many AI and OTCS standards.

5 Discussion

It appears the cybersecurity threat models for machine learning, both theoretical and practical, have already emerged, but there is still too little information available to verify if attacks against ML components in OT or logistics applications follow these new patterns.

Standardization of ML and OT cybersecurity advances rapidly on both tracks independently, but there seems to be little interconnection between those tracks. There is, for example, a recent and wide coverage of OTCS standards in each logistics sector. However, the recent updates or completely new standards also indicate that there has not been time for applications and industries to adopt them.

Cyberattacks and other offensive cyber activity are shifting slowly toward OT and ML implementations as well. However, making ML implementations resistant to different kinds of manipulation efforts seems to make them fragile if they are given partly conflicting security objectives, such as security and safety. This fragility will imply a challenge to the future implementations in safety-critical application in logistics.

The safety-critical property in many OT systems is often thought to require absolute certainty of the absence of unsafe states from the system. However, this may not be possible to attain with ML implementations with exponentially larger state-universe. It may then be necessary to move to dynamic risk management models, if certain types of ML are used in safety-critical OT applications.

Different AI and ML implementations outside ANNs have existed for some decades already and found their niche areas in OT as well. Some of these older techniques are easily verified and understood due to their simplicity and static nature during the inference phase. This is actually an advantage in safety-critical logistics applications.

Based on the number of recent cyber-incidents,[6] ML or OT systems in logistics are not under a major threat: the overwhelming majority of cyberattacks use offensive cyber methods against conventional IT systems rather than ML or OT systems.

The future of OT and logistics IT in general will undoubtedly involve increasing use of AI and ML. It is therefore important to acknowledge the specific characteristics of the application domain, for example, one defining attribute of the applications in the domain is their federation. Federated ML techniques will play a large role in logistics, and the security techniques used there need to be expanded from federated learning to other phases of the AI systems and data life cycle.

6 Conclusions

In this survey, we set out to map the landscape in the intersection of cybersecurity threats, machine learning, and logistics-related OT. Our null hypothesis was that this is still very much unchartered territory. There is indeed some research already available, but especially sector-specific applied research is very scarce.

The main findings about the status quo of cybersecurity of ML in OT and logistics applications include, for example, that most known attack types directed particularly at compromising ML components are designed against ANNs. However, logistics applications have longer life cycles, and older AI types still cover a significant percentage of implementations in most domains and modes. Thus, in the current cyber threat landscape, many ML implementations in logistics are "under the radar." This should not be taken as a security measure, though.

The existing OT cybersecurity standards do not address ML at all or at most recommend using ML-IDS for cybersecurity. The landscape is changing, though: authorities are rolling out general AI cybersecurity standards and guidance with increasing speed. However, OT or logistics specific (AI cybersecurity) guidance is slower to emerge. Some development can be seen in the aviation sector, one example being future standard EUROCAE ED324 [93].

There is also a definite lack of applied research of the cyber resiliency of some important logistics ML applications in predictive and optimization models and even some blind spots, such as smart port analytics.

Even though the focus area in this survey was the intersection of OT, ML, and logistics cybersecurity, the attack surface of the combination is actually

[6] By October 2023 as of the time of this writing

multiplicative: combining IT, OT, and ML system and data domains will increase the attack surface sixfold. Attacks using almost all of these partial attack surfaces have already been tested (e.g., in red-teaming exercises against critical infrastructure OT).

References

1. European Union Agency for Network and Information Security (ENISA): Cyber Security of AI and Standardisation. ISBN 978-92-9204-616-3, https://doi.org/10.2824/277479 (2023)
2. Parkinson, H., Basher, D., Bamford, G.: Railway cyber security and TS50701. https://doi.org/10.4203/ccc.1.17.1 (2022)
3. Mission Secure: A Comprehensive Guide to Maritime Cybersecurity. ebook. https://www.missionsecure.com/maritime-security-perspectives-for-a-comprehensive-approach
4. European Commission: Proposal for a regulation of the European Parliament and of the Council laying down harmonized rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. Ch.5.2.3, Explanatory Memorandum, 21.4.2021
5. IATA.: Compilation of cyber security regulations, standards, and guidance applicable to civil aviation. Ed.3.0, Dec/2021. https://www.iata.org/contentassets/4c51b00fb25e4b60b38376a4935e278b/compilation-of-cyber-regulations-standards-and-guidance_3.0.pdf
6. EUROCONTROL: EATM-CERT aviation cyber events map – Google My Maps. https://www.eurocontrol.int/cybersecurity
7. ISO: Definition of Operational Technology (OT). ISO/IEC JTC 1/SC 27 CfC on Operational Technology (n.o 23088), 17.7.2023
8. NIST, US DoC: NIST AI 100-2e2023 ipd: Adversarial Machine Learning, A Taxonomy and Terminology of Attacks and Mitigations (pp. 11). https://doi.org/10.6028/NIST.AI.100-2e2023.ipd, March 2023
9. NIST: Underlying Technical Models for Information Technology Security, NIST Special Publication 800-33. Recommendations of the National Institute of Standards and Technology (retired). https://doi.org/10.6028/NIST.SP.800-33, December 2001
10. European Union: Regulation (EU) 2023/1230 of the European Parliament and of the Council of 14 June 2023 on machinery and repealing Directive 2006/42/EC of the European Parliament and of the Council and Council Directive 73/361/EEC. EurLEX document 32023R1230, https://eur-lex.europa.eu/eli/reg/2023/1230/oj, 29.6.2023
11. European Commission, Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs: Impact assessment study on the revision of Directive 2006/42/EC on machinery. Publications Office, 2020, https://data.europa.eu/doi/10.2873/423938
12. Wurldtech: An Executive Guide to Cyber Security for Operational Technology: Securing critical assets in a digitally connected world. https://scadahacker.com/library/Documents/White_Papers/Wurldtech%20-%20An%20Executive%20Guide%20to%20Cyber%20Security%20for%20Operational%20Technology.pdf (2016)
13. Eggers, S.: Towards a New Supply Chain Cybersecurity Risk Analysis Technique. FY21 DOE-NE Cybersecurity Supply Chain Research Report, INL/EXT-21-64089, Rev.0, Idaho National Laboratory. Aug/2021
14. Zizzo, G., Hankin, C., Maffeis, S., Jones, K.: Intrusion detection for industrial control systems: evaluation analysis and adversarial attacks. arXiv preprint: 1911.04278 (2019)
15. Ginter, A., Hale, G., Machtemes, R., Molina, J., Wallhof, M., Schneider, C.: 2023 Threat Report – OT Cyberattacks with Physical Consequences. https://waterfall-security.com/ot-insights-center/ot-cybersecurity-insights-center/2023-threat-report-ot-cyberattacks-with-physical-consequences/ (2023)

16. Microsoft, MITRE: Microsoft Azure Service Disruption – Exercise. A MITRE ATLAS Case Study n.o. CS0010. https://atlas.mitre.org/studies/AML.CS0010 (2020)
17. Responsible AI Collaborative: AIID, Artificial Intelligence Incident Database. https://incidentdatabase.ai/
18. MITRE Corp.: MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems). https://atlas.mitre.org/, Sep/2023
19. MITRE Corp.: MITRE ATT&CK (Adversarial Tactics, Techniques and Common Knowledge). https://attack.mitre.org/, Sep/2023
20. MITRE Corp.: MITRE D3FEND (Adversarial Threat Landscape for Artificial-Intelligence Systems). https://d3fend.mitre.org/, Sep/2023
21. Tidjon, L.N., Khomh, F.: Threat Assessment in Machine Learning based Systems. arXiv:2207.00091. https://arxiv.org/abs/2207.00091, 30.7.2022
22. ISS Source, Adolus, Waterfall Inc.: Industrial Control System ICS STRIVE Covering security, threats, regulations, incidents, vulnerabilities with experts. https://icsstrive.com/?wpv-industry%5B%5D=aerospace&wpv-industry%5B%5D=automotive&wpv-industry%5B%5D=transportation&wpv_aux_current_post_id=153&wpv_aux_parent_post_id=153&wpv_view_count=9385. Accessed 24 Sept 2023
23. Meland, P.H., Bernsmed, K., Wille, E., Rødseth, Ø.J., Nesheim, D.A.: A Retrospective Analysis of Maritime Cyber Security Incidents. TransNav Int. J. Mar. Navig. Safety Sea Transport. **15**(3), 519–530 (2021). https://doi.org/10.12716/1001.15.03.04
24. ISO: ISO/IEC TR 24030:2021, Information technology — Artificial intelligence (AI) — Use cases. https://www.iso.org/standard/77610.html (2021)
25. Akbari, W., Anh Do, T.N.: A systematic review of machine learning in logistics and supply chain management: current trends and future directions. Benchmarking Int. J. **5**, 11 (2021)
26. Singh, A., Wiktorsson, M., Hauge, J.B.: Trends in machine learning to solve problems in logistics
27. Tsolaki, K., et al.: Utilizing machine learning on freight transportation and logistics applications: a review. ICT Exp. **10**, 2 (2022)
28. LF AI Foundation: Adversarial Robustness Toolbox. Trusted and Responsible AI, attacks. https://github.com/Trusted-AI/adversarial-robustness-toolbox
29. ENISA: Port Cybersecurity: Good practices for cybersecurity in the maritime sector. https://doi.org/10.2824/328515, Nov/2019
30. NIST: NIST SP 800-82 Rev. 3: Guide to Operational Technology (OT) Security. NIST CSRC Publications. https://doi.org/10.6028/NIST.SP.800-82r3, Sep/2023.
31. Gheyas, I., Epiphaniou, G., Maple, C., Lakshminarayana, S.: A resilient cyber-physical demand forecasting system for critical infrastructures against stealthy false data injection attacks. Appl. Sci. **12**, 10093 (2022). https://doi.org/10.3390/app121910093
32. Chen, J., Gao, Y., Shan, J., Peng, K., Wang, C., Jiang, H.: Manipulating supply chain demand forecasting with targeted poisoning attacks. IEEE Trans. Ind. Inform. **19**, 1803–1813 (2023). https://doi.org/10.1109/TII.2022.3175958
33. Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., Wang, K.I.-K.: Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet Things J. **9**(12), 9310–9319 (2022). https://doi.org/10.1109/JIOT.2021.3130434
34. Zhu, L., Feng, K., Pu, Z., Ma, W.: Adversarial diffusion attacks on graph-based traffic prediction models. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2023.3290401
35. Mode, G.R., Hoque, K.A.: Crafting adversarial examples for deep learning based prognostics (extended version). arXiv:2009.10149 (2020)
36. Gungor, O., Rosing, T., Aksanli, B.: STEWART: STacking Ensemble for White-Box AdversaRial Attacks Towards more resilient data-driven predictive maintenance. Comput. Ind. **140** (2022), 103660, ISSN 0166-3615. https://doi.org/10.1016/j.compind.2022.103660
37. Eirini, A., Williams, L., Rhode, M., Burnap, P., Wedgbury, A.: Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems. Journal of Information Security and Applications. **58**, 102717 (2021). https://doi.org/10.1016/j.jisa.2020.102717

38. Mode, G., Calyam, P., Hoque, K.: Impact of false data injection attacks on deep learning enabled predictive analytics. https://doi.org/10.1109/NOMS47738.2020.9110395 (2020)
39. Mulo, J., Tian, P., Hussaini, A., Liang, H., Yu, W.: Towards an adversarial machine learning framework in cyber-physical systems. 2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA), Orlando, FL, USA, 2023, pp. 138–143. https://doi.org/10.1109/SERA57763.2023.10197774
40. Maritime gateway: Port of Rotterdam targeted in cyberattack. https://www.maritimegateway.com/port-of-rotterdam-targeted-in-cyberattack/, 16.6.2023
41. Liu, T., Xu, X., Lei, Z., Zhang, X., Sha, M., Wang, F.: A multi-task deep learning model integrating ship trajectory and collision risk prediction. Ocean Eng. **287**(Part 2), 115870., ISSN 0029-8018, (2023). https://doi.org/10.1016/j.oceaneng.2023.115870
42. Filom, S., Amiri, A.M., Razavi, S.: Applications of machine learning methods in port operations – a systematic literature review. Transp. Res. E Logist. Transp. Rev. **161**, 102722., ISSN 1366-5545 (2022). https://doi.org/10.1016/j.tre.2022.102722
43. Tirkolaee, E.B., Darvazeh, S., Farzaneh, M., Vandchali, R., Samira, A.: Application of machine learning in supply chain management: a comprehensive overview of the main areas. Math. Probl. Eng., 1–14 (2021). https://doi.org/10.1155/2021/1476043
44. da Silveira Dib, M., Prates, P., Ribeiro, B.: SecFL – Secure Federated Learning Framework for predicting defects in sheet metal forming under variability. Expert Syst. Appl. **235**, 121139., ISSN 0957-4174 (2024). https://doi.org/10.1016/j.eswa.2023.121139
45. Liu, G., Lai, L.: Provably efficient black-box action poisoning attacks against reinforcement learning. In: 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia. https://arxiv.org/abs/2110.04471v2 (2021)
46. Neelen, W., van Duijn, R.: Hacking traffic lights. Defcon. (2020) https://media.defcon.org/DEF%20CON%2028/DEF%20CON%20Safe%20Mode%20presentations/DEF%20CON%20Safe%20Mode%20-%20Wesley%20Neelen%20%26%20Rik%20van%20Duijn%20-%20Hacking%20Traffic%20Lights.pdf
47. Holroyd, M.: Euronews. Gridlock as hackers order hundreds of taxis to same place in Moscow. https://www.euronews.com/my-europe/2022/09/02/gridlock-as-hackers-order-hundreds-of-taxis-to-same-place-in-moscow (2022)
48. Cao, H., Zou, W., Wang, Y., Song, T., Liu, M. Emerging threats in deep learning-based autonomous driving: a comprehensive survey. https://arxiv.org/pdf/2210.11237v1.pdf
49. M. Cosic (Thisismoney.co.uk): Criminals could hack into self-driving cars to launch terror attacks or carry out 'cash for crash' frauds, insurers warn. https://www.thisismoney.co.uk/money/cars/article-11998451/Criminals-hack-self-driving-cars-launch-terror-attacks-frauds-insurers-warn.html, 21.4.2023
50. S. Calder (Indepent): Cyber attacks and dozy drivers: these are the future risks of self-driving cars. https://www.independent.co.uk/travel/news-and-advice/autonomous-vehicles-cyber-attacks-danger-b2411929.html, 17.9.2023
51. Chen, Y., Zhu, X., Gong, X., Yi, X., Li, S.: Data poisoning attacks in internet-of-vehicle networks: Taxonomy, state-of-the-art, and future directions. IEEE Trans. Ind. Inform., 1–9 (2022)
52. Talpur, A., Gurusamy, M.: GFCL: A GRU-based Federated Continual Learning Framework against Data Poisoning Attacks in IoV. https://doi.org/10.48550/arXiv.2204.11010, 12.9.2022
53. Wang, F., Wang, X., Hong, Y., Ban, X.: Data Poisoning Attacks on Traffic State Estimation and Prediction (TSEP). https://doi.org/10.2139/ssrn.4396123, 5.12.2022
54. MITRE ATLAS: Compromised PyTorch Dependency Chain. Case Study AML.CS0015 (incident). https://atlas.mitre.org/studies/AML.CS0015, 25.12.2022
55. Liang, B., Su, M., You, W., Shi, W., Yang, G.: Cracking classifiers for evasion: a case study on the Google's phishing pages filter. Procs of WWW, 345–356 (2016). https://doi.org/10.1145/2872427.2883060

56. Zakariyya, I., Kalutarage, H., Al-Kadri, M.: Towards a robust, effective and resource efficient machine learning technique for IoT security monitoring. Comput. Secur. **133**, 103388 (2023). https://doi.org/10.1016/j.cose.2023.103388
57. Li, J.: Towards Secure Deep Neural Networks for Cyber-Physical Systems. A Dissertation Presented for the Doctor of Philosophy Degree, The University of Tennessee, Knoxville. https://trace.tennessee.edu/cgi/viewcontent.cgi?article=7815&context=utk_graddiss, May/2021
58. Heidari, A., Navimipour, N., Unal, M., Zhang, G.: Machine Learning Applications in Internet-of-Drones: Systematic Review, Recent Deployments, and Open Issues. ACM Comput. Surv. **55**(12) Article 247, 45 (2023). https://doi.org/10.1145/3571728
59. Ding, Y., Yang, Z., Pham, Q-V., Zhang, Z., Shikh-Bahaei, M.: Distributed Machine Learning for UAV Swarms: Computing, Sensing, and Semantics. https://arxiv.org/pdf/2301.00912v1.pdf, 3.1.2023
60. Davidson, D., Wu, H., Jellinek, R., Singh, V., Ristenpart, T.: Controlling UAVs with sensor input spoofing attacks. In: 10th {USENIX} Workshop on Offensive Technologies ({WOOT} 16) (2016)
61. Kim, K., Nalluri, S., Kashinath, A., Wang, Y., Mohan, S., Miroslav, P., Bo, L.: Security analysis against spoofing attacks for distributed UAVs. In: Workshop on Decentralized IoT Systems and Security (DISS) 2020. https://doi.org/10.14722/diss.2020.23011, 23.2.2020
62. Walter, M.J., Barrett, A., Walker, D.J., Tam, K.: Adversarial AI testcases for maritime autonomous systems. In: AI, Computer Science and Robotics Technology, vol. 2. IntechOpen (2023). https://doi.org/10.5772/acrt.15
63. Lee, C., Lee, S.: Vulnerability of clean-label poisoning attack for object detection. Maritime Auton. Surface Ships J. Mar. Sci. Eng. **11**, 1179 (2023). https://doi.org/10.3390/jmse11061179
64. Singh, P., Dulebenets, M.A., Pasha, J., Gonzalez, E.D.R.S., Lau, Y.-Y., Kampmann, R.: Deployment of autonomous trains in rail transportation: current trends and existing challenges. IEEE Access. **9**, 91427–91461 (2021). https://doi.org/10.1109/ACCESS.2021.3091550
65. Tong, C., Jiqiang, L., Yingxiao, X., et al.: Adversarial retraining attack of asynchronous advantage actor-critic based pathfinding. Int. J. Intell. Syst. **36**, 2323–2346 (2021). https://doi.org/10.1002/int.22380
66. Chen, T., Liu, J.Q., Li, H., et al.: Robustness assessment of asynchronous advantage actor-critic based on dynamic skewness and sparseness computation: a parallel computing view. J. Comput. Sci. Technol. **36**, 1002–1021 (2021). https://doi.org/10.1007/s11390-021-1217-z
67. Ilascu, I. (Bleeping Computer): New details emerge of Fxmsp's hacking of antivirus companies. Bleeping Computer Security News. https://www.bleepingcomputer.com/news/security/new-details-emerge-of-fxmsps-hacking-of-antivirus-companies/, 13.5.2019
68. Kovacs, E. (Securityweek). Ransomware Gang Leaks Files Stolen from Industrial Giant Parker Hannifin. https://www.securityweek.com/ransomware-gang-leaks-files-stolen-industrial-giant-parker-hannifin/, 5.4.2022
69. Wang, P., Li, Y., Shekhar, S., Northrop, W.F.: Adversarial attacks on reinforcement learning based energy management systems of extended range electric delivery vehicles. ArXiv, abs/2006.00817 (2020)
70. Soderi, S., Masti, D., Lun, Y.Z.: Railway cyber-security in the era of interconnected systems: a survey. IEEE Trans. Intell. Transp. Syst. **24**(7) (2023). https://doi.org/10.1109/TITS.2023.3254442
71. ENISA: ENISA Threat Landscape: Transport Sector (January 2021 to October 2022). https://www.enisa.europa.eu/publications/enisa-transport-threat-landscape, 3/2023
72. ENISA: ENISA Threat Landscape for Supply Chain Attacks. https://www.enisa.europa.eu/publications/threat-landscape-for-supply-chain-attacks, 7/2021.
73. Rail Industry System and Standards Board (RISSB): AS 7770:2018 – Rail Cyber Security. Australian standard. https://www.rissb.com.au/products/as-7770-rail-cyber-security/ (2018)

74. International Society of Automation (ISA): ISA/IEC 62443 Series of Standards. https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards
75. VDE Verlag: Electric signalling systems for railways, Part 104: IT Security Guideline based on IEC 62443", DIN VDE V 0831-104:2015-10. https://www.vde-verlag.de/standards/0800264/din-vde-v-0831-104-vde-v-0831-104-2015-10.html, Oct/2015
76. CENELEC: Railway applications – Cybersecurity. CLC/TS 50701:2023. https://standards.cencenelec.eu/dyn/www/f?p=CENELEC:110:0::::FSP_PROJECT:74651&cs=1C40E3012B9331CD5D71A44786D838F8C, 11.8.2023
77. CYRail Consortium: CYRail recommendations on cybersecurity of rail signalling and communication system. Deliverable 7.5/EU Horizon 2020 project: CYbersecurity in the RAILway sector. https://cyrail.eu/IMG/pdf/final_recommendations_cyrail.pdf, Sep/2018
78. International Union of Railways (IUC) ARGUS WG: Guidelines for cyber-security in railway. ISBN: 978-2-7461-2732-6. https://shop.uic.org/en/other-documents/9228-guidelines-for-cyber-security-in-railways.html, Jun/2018
79. ENISA: Railway cybersecurity, Good practices in cyber risk management. https://www.enisa.europa.eu/publications/railway-cybersecurity, 13.11.2020
80. International Electrotechnical Commission (IEC): IEC 63154:2021 Maritime navigation and radiocommunication equipment and systems – Cybersecurity – General requirements, methods of testing and required test results. https://webstore.iec.ch/publication/61003, 9.3.2021.
81. Baltic and International Maritime Council (BIMCO): The Guidelines on Cyber Security Onboard Ships, Version 4. https://www.bimco.org/about-us-and-our-members/publications/the-guidelines-on-cyber-security-onboard-ships (2018)
82. International Maritime Organization (IMO). Guidelines on Maritime Cyber Risk Management. MSC-FAL.1/Circ.3. https://www.imca-int.com/information-notes/imo-guidelines-maritime-cyber-risk-management/, 16.8.2017
83. International Association of Classification Societies (IACS): Recommendation on Cyber Resilience. https://iacs.org.uk/resolutions/recommendations/161-180/rec-166-new-corr2-cln, Rec. no.166. 2.4.2022.
84. International Association of Classification Societies (IACS): Cyber resilience of ships, Unified requirement E26, (withdrawn from iacs.org.uk, new version pending as of the time of writing), Apr/2022
85. International Association of Classification Societies (IACS): Cyber resilience of on-board systems and equipment. Unified requirement E27, (withdrawn from iacs.org.uk, new version pending as of the time of writing), Apr/2022
86. DNV GL AS.: Cyber security resilience management for ships and mobile offshore units in operation. DNV GL recommended practice DNVGL-RP-0496. https://standards.dnv.com/explorer/document/0ED73B3209DA42CDA6392BC3946585C9/4, Sep/2016
87. EU Aviation Safety Agency (EASA): Executive Director Decision 2020/006/R. https://www.easa.europa.eu/en/document-library/agency-decisions/ed-decision-2020006r, 1.7.2020
88. EU Aviation Safety Agency (EASA): AMC/GM to Part 21 — Issue 2, Amendment 10. Annex to ED Decision 2020/006/R, https://www.easa.europa.eu/en/downloads/116277/en, 1.7.2020
89. European Organisation for Civil Aviation Equipment (EUROCAE): ED-202A – Airworthiness Security Process Specification. https://eshop.eurocae.net/eurocae-documents-and-reports/ed-202a/, June/2014
90. European Organisation for Civil Aviation Equipment (EUROCAE): ED-203A – Airworthiness Security Methods and Considerations. https://eshop.eurocae.net/eurocae-documents-and-reports/ed-203a/, June/2018
91. European Organisation for Civil Aviation Equipment (EUROCAE): ED-204A – Information Security Guidance for Continuing Airworthiness. https://eshop.eurocae.net/eurocae-documents-and-reports/ed-204a-information-security-guidance-for-continuing-airworthiness/, Sep/2020
92. European Organisation for Civil Aviation Equipment (EUROCAE): ED-205A – Process Standard for Security Certification and Declaration of ATM ANS Ground Systems. https://eshop.eurocae.net/eurocae-documents-and-reports/ed-205a/, Jul/2022

93. European Organisation for Civil Aviation Equipment (EUROCAE): WG-114/Artificial Intelligence. https://eurocae.net/about-us/working-groups/ (2023)
94. SAE International; Aircraft Data Network, Part 1: Systems Concepts and Overview: ARINC664P1-1. https://www.sae.org/standards/content/arinc664p1-1/, 30.6.2006
95. SAE International: Datalink Security, part 1 - ACARS message security: ARINC823P1. https://www.sae.org/standards/content/arinc823p1/, 12.10.2007
96. SAE International: Aircraft Data Interface Function (ADIF): ARINC834-8. https://www.sae.org/standards/content/arinc834-8/, 21.7.2020.
97. SAE International: Guidance for Security of Loadable Software Parts Using Digital Signatures: ARINC835-1. https://www.sae.org/standards/content/arinc835-1/, 2.1.2014
98. SAE International. Guidance for Security Event Logging in an IP-environment: ARINC852. https://www.sae.org/standards/content/arinc852/, 21.6.2017
99. SAE International: Internet Protocol Suite (IPS) for Aeronautical Safety Services Part 1: Airborne IPS System Technical Requirements: ARINC858P1. https://www.sae.org/standards/content/arinc858p1/, 21.6.2021
100. Airlines for America (A4A): Spec 42: Aviation Industry Standards for Digital Information Security. Rev. 2020.1. https://publications.airlines.org/CommerceProductDetail.aspx?Product=294, Jan/2020
101. Eggers, S., Idaho National Laboratory: Towards a New Supply Chain Cybersecurity Risk Analysis Technique. INL/EXT-21-64089, Rev. 0, FY21 DOE-NE Cybersecurity Supply Chain Research Report. https://inldigitallibrary.inl.gov/sites/sti/sti/Sort_50869.pdf, Aug/2021
102. SAE International: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems: J3061_202112. https://www.sae.org/standards/content/j3061_202112/, 15.12.2021
103. SAE International: Road Vehicles – Cybersecurity Engineering: ISO/SAE21434. https://www.sae.org/standards/content/iso/sae21434/, 31.8.2021
104. International Standardization Organization (ISO): ISO/TS 21177:2023, Intelligent transport systems – ITS station security services for secure session establishment and authentication between trusted devices. https://www.iso.org/standard/81067.html, Apr/2023
105. United Nations Economic Commission for Europe (UNECE): UN Regulation No. 155 – Cyber security and cyber security management system. E/ECE/TRANS/505/Rev.3/Add.154, https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security, 4.3.2021
106. National Highway Traffic Safety Administration (NHTSA): Cybersecurity Best Practices for the Safety of Modern Vehicles | 2022. https://www.nhtsa.gov/document/cybersecurity-best-practices-safety-modern-vehicles-2022, Sep/2022
107. AUTOSAR: AUTOSAR Adaptive Platform. AUTOSAR Adaptive Release R22-11. https://www.autosar.org/standards/adaptive-platform, Nov/2022
108. NIST: The NIST Cybersecurity Framework 2.0. NIST CSWP 29 (Initial Public Draft), NIST CSRC Publications. https://doi.org/10.6028/NIST.CSWP.29.ipd, Aug/2023
109. International Standardization Organization (ISO): ISO 28004-2:2014, Security management systems for the supply chain — Guidelines for the implementation of ISO 28000 — Part 2: Guidelines for adopting ISO 28000 for use in medium and small seaport operations. https://www.iso.org/standard/60905.html, Feb/2014
110. International Standardization Organization (ISO): ISO/IEC 20243-1:2018 Information technology — Open Trusted Technology ProviderTM Standard (O-TTPS) — Mitigating maliciously tainted and counterfeit products — Part 1: Requirements and recommendations. https://www.iso.org/standard/74399.html, Feb/2018
111. SAE International: Counterfeit Electrical, Electronic, and Electromechanical (EEE) Parts; Avoidance, Detection, Mitigation, and Disposition: AS5553D. https://www.sae.org/standards/content/as5553d/, 14.4.2022
112. SAE International: Counterfeit Electrical, Electronic, and Electromechanical (EEE) Parts: Avoidance, Detection, Mitigation, and Disposition – Independent Distribution: AS6081A. https://www.sae.org/standards/content/as6081a/, 21.4.2023

113. SAE International: Counterfeit Electrical, Electronic, and Electromechanical (EEE) Parts: Tools for Risk Assessment of Other than an Authorized Source (e.g., Independent Distributors) ARP6178A. https://www.sae.org/standards/content/arp6178a/, 1.8.2023
114. SAE International: Supply Chain Risk Management Guideline: ARP9134A. https://www.sae.org/standards/content/arp9134a/, 6.2.2014
115. UL Solutions LLC. Outline Of Investigation For Software Cybersecurity For Network-Connectable Products, Part 2-2: Particular Requirements For Industrial Control Systems. UL 2900-2-2 Ed. 1-2016, https://webstore.ansi.org/standards/ul/ul2900ed2016-1660595 (2016)
116. European Telecommunications Standards Institute (ETSI): Securing Artificial Intelligence (SAI); Artificial Intelligence Computing Platform Security Framework. ETSI GR SAI 009 V1.1.1 (2023-02), Group Report. https://www.etsi.org/deliver/etsi_gr/SAI/001_099/009/01.01.01_60/gr_SAI009v010101p.pdf, Feb/2023
117. European Telecommunications Standards Institute (ETSI): SmartM2M; Artificial Intelligence and the oneM2M architecture. ETSI TR 103 674 V1.1.1 (2021-02), Technical report. https://www.etsi.org/deliver/etsi_tr/103600_103699/103674/01.01.01_60/tr_103674v010101p.pdf, Feb/2021
118. NIST: Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1, NIST AIRC publicatons. https://doi.org/10.6028/NIST.AI.100-1, Jan/2023
119. NIST: The Language of Trustworthy AI: An In-Depth Glossary of Terms. NIST AI 100-3, NIST AIRC publicatons. https://doi.org/10.6028/NIST.AI.100-3, Mar/2023
120. DKE German Commission for Electrical, Electronic & Information Technologies of DIN and VDE: German Standardization Roadmap on Artificial Intelligence. https://www.din.de/resource/blob/772610/e96c34dd6b12900ea75b460538805349/normungsroadmap-en-data.pdf, Nov/2020
121. International Standardization Organization (ISO): ISO/IEC 20547-4:2020 Information technology – Big data reference architecture – Part 4: Security and privacy. https://www.iso.org/standard/71278.html, Sep/2020
122. International Standardization Organization (ISO): ISO/IEC 23894:2023 Information technology – Artificial intelligence – Guidance on risk management. https://www.iso.org/standard/77304.html. Feb/2023
123. International Standardization Organization (ISO): ISO/IEC TR 24028:2020 Information technology – Artificial intelligence – Overview of trustworthiness in artificial intelligence. https://www.iso.org/standard/77608.html, May/2020
124. International Standardization Organization (ISO): ISO/IEC TR 24029-1:2021 Artificial Intelligence (AI) – Assessment of the robustness of neural networks – Part 1: Overview. https://www.iso.org/standard/77609.html, Mar/2021
125. International Standardization Organization (ISO): ISO/IEC TR 27563:2023 Security and privacy in artificial intelligence use cases – Best practices. https://www.iso.org/standard/80396.html, May/2023
126. International Standardization Organization (ISO): ISO/IEC TR 29119-11:2020 Software and systems engineering – Software testing – Part 11: Guidelines on the testing of AI-based systems. https://www.iso.org/standard/79016.html, Nov/2020
127. Traficom, Liikenne ja viestintävirasto: Recommendation for Promoting Cyber Security in Rail Transport. TRAFICOM/517957/03.04.02.01/2022. https://www.traficom.fi/fi/saadokset/suositus-kyberturvallisuuden-edistamisesta-raideliikenteessa, 30.1.2023
128. Bundesamt für Sicherheit in der Informationstechnik (BSI): Security of AI-Systems: Fundamentals, Adversarial Deep Learning. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/KI/Security-of-AI-systems_fundamentals.html, 15.8.2022

Fuzzy Machine Learning for Smart Grid Instability Detection

Fabio Martinelli, Francesco Mercaldo, and Antonella Santone

1 Introduction and Related Work

The rise of renewable energy sources offers a highly sought-after alternative to finite and environmentally harmful fossil fuels, meeting the global community's demands. However, embracing these renewable sources presents new paradigms and research challenges, for instance, before the emergence of renewable energy, the traditional operating ecosystem involved only a few energy production entities supplying power to consumers through unidirectional flows [20]. Indeed, with the introduction of renewable energy options, end users, including households and enterprises, now have the capability not only to consume energy but also to produce and supply it. These individuals are commonly referred to as "prosumers," as they act as both producers and consumers of energy [4]. Consequently, energy flow within smart grids has shifted to become bidirectional: the significant potential for flexibility within smart grids, which has been made possible by the adoption of renewable sources and the emergence of "prosumers." However, this shift has also brought forth greater complexity in managing energy supply and demand within the system, leading to more challenging economic implications [19].

F. Martinelli (✉)
Institute for Informatics and Telematics, National Research Council of Italy (CNR), Pisa, Italy
e-mail: fabio.martinelli@iit.cnr.it

F. Mercaldo
Institute for Informatics and Telematics, National Research Council of Italy (CNR), Pisa, Italy

University of Molise, Campobasso, Italy
e-mail: francesco.mercaldo@iit.cnr.it; francesco.mercaldo@unimol.it

A. Santone
University of Molise, Campobasso, Italy
e-mail: antonella.santone@unimol.i

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_10

Relevant contributions on how to tackle the requirements of such new scenario have been offered by academy and industry over the past years [15, 16]. In the smart grid context, great interest has been devoted to the study of smart grid stability [8, 12, 19].

Indeed, within a smart grid, data on consumer demand is gathered and centrally assessed in relation to the present supply conditions. Based on this evaluation, price information is generated and sent back to customers, empowering them to make informed decisions regarding their energy usage. Due to the time-sensitive nature of this entire process, the dynamic estimation of grid stability becomes not just a matter of concern but a crucial necessity [3].

Simply speaking, the idea is to understand and plan for both energy production and/or consumption disturbances and fluctuations introduced by system participants in a dynamic way [13], taking into consideration not only technical aspects but also how participants respond to changes in the associated economic aspects (i.e., energy price) [17].

For the stability of a system with a smart grid, there are two main criteria to respect [12]: first, the generation has to match the demand at any time and has to hold a reserve (battery storage) for immediate outages. Second, the grid has to provide sufficient capacity for voltage stability at every smart grid portion.

Considering the importance to detect instability situations in smart grids, in this paper we propose a method aimed to predict in real-time the stability of a smart grid. We consider fuzzy machine learning models by authors by taking into account also a kind of explainability, aimed to understand how the model is working from a global point of view.

State-of-the-art research includes proposals to detect stability in smart grids. For instance, authors in [6] exploit extreme machine learning to predict smart grid stability by considering a dataset composed by 10,000 stable and unstable smart grid observations, by obtaining an accuracy equal to 0.98.

Researchers in [7] evaluated the effectiveness of the Extreme Gradient Boosting model for smart grid instability detection by reaching an accuracy of 0.95.

Zare and colleagues [22] design and developed an adaptive scheme for predicting out-of-step (OOS) condition of synchronous generator based on the Bayesian technique. For classifying target classes between stable and OOS conditions, a series of measurements are derived under various fault scenarios including topological and operational disturbances. The proposed approach is applied on IEEE 39-bus test system from which by using trained variables, the tripping signals are estimated online. They reach an accuracy equal to 91.6%.

Aghamohammadi et al. [1] for early prediction of OOS condition of an unstable generator, propose a three-stage decision tree-based approach consisting of fault detector DT (FDDT), clearance detector DT (CDDT), and instability predictor DT (IPDT). The proposed algorithm is demonstrated on generators of IEEE 39-bus test system, obtaining an accuracy equal to 90.3%.

Gupta and colleagues [9] adopt a convolutional neural network, whose input is the heatmap representation of the measurements, for instability prediction. They propose a continuous Online Monitoring System (OMS) for assessment of rotor

angle stability in a power system. The OMS uses voltage magnitudes and voltage angles at all the generator buses as inputs, which are usually available from phasor (PMU) measurements. An accuracy of 89.22% is obtained by experimenting the proposed method on IEEE 118-bus and IEEE 145-bus systems.

Arzamasov and colleagues [3] apply decision trees with the aim to discover situations of instability: they obtain an accuracy of 80% exploiting a four-node star architecture.

With the Bagging Classifier, Tiwari et al. [21] obtained an accuracy equal to 0.97, while XGBoost classifier obtained an accuracy of 0.97 in reference [8]. Moldovan and colleagues [18] reached an accuracy equal to 0.93 with the multilayer perceptron classification algorithm. An accuracy equal to 0.99 is reached by deep learning models that are exploited by Breviglieri and colleagues [4]

As highlighted by the state-of-the-art literature, researchers basically exploit artificial intelligence for smart grid stability detection.

The main difference between the discussed papers and the proposed method is the adoption of fuzzy machine learning algorithms for smart grid stability detection.

Fuzzy logic and artificial intelligence (AI) are two distinct but interconnected concepts within the field of computer science and decision-making systems.

Fuzzy logic is a branch of mathematics and a formal logic system designed to handle uncertainties and imprecision in decision-making. Unlike traditional binary logic (Boolean logic), which uses crisp values of true (1) and false (0), fuzzy logic allows for intermediate values, which are expressed as degrees of truth between 0 and 1. These intermediate values represent the degree of membership of an element in a set.

Fuzzy logic is particularly useful when dealing with vague or ambiguous concepts, as it can handle linguistic terms and human-like reasoning. It finds applications in various fields, including control systems, expert systems, pattern recognition, and decision support systems.

Artificial intelligence is a broad area of computer science that focuses on creating machines or computer programs capable of performing tasks that typically require human intelligence. AI systems can learn from data, reason, make decisions, and solve problems, often mimicking human cognitive functions.

AI encompasses various subfields, such as machine learning, natural language processing, computer vision, robotics, and more. These AI techniques can be used in a wide range of applications, including autonomous vehicles, virtual assistants, fraud detection, healthcare diagnosis, and recommendation systems.

Fuzzy logic and AI can be intertwined in certain AI systems, especially in decision-making and control scenarios. Fuzzy logic can be used to model and manage uncertainty in AI algorithms, making them more robust and adaptable to real-world complexities. For example, in an autonomous vehicle's control system, fuzzy logic can be applied to handle imprecise sensor data or ambiguous environmental conditions.

Furthermore, fuzzy logic can be integrated with machine learning algorithms to improve the interpretability of AI models. While some AI algorithms may provide accurate predictions, they might lack transparency in explaining the reasoning

behind their decisions. Fuzzy logic can add a layer of interpretability by providing linguistic rules that human operators can understand, making the AI system more trustworthy and explainable.

In summary, fuzzy logic and AI are both essential components of modern intelligent systems, and their combination can enhance the capabilities and performance of AI algorithms, particularly in scenarios involving uncertainty and imprecision.

The paper proceeds as follows: in the next section, we present the proposed method, in Sect. 3 the results of the experimental analysis are discussed, and, finally, in the last section, conclusion and future research lines are drawn.

2 Fuzzy Machine Learning for Smart Grid State Detection

In the following section, the proposed method aimed to detect stability and instability in smart grids by exploiting fuzzy machine learning classification algorithms is presented.

Fuzzy machine learning is a specialized area that combines concepts from fuzzy logic and machine learning. It aims to enhance traditional machine learning algorithms by incorporating fuzzy sets and fuzzy logic principles to deal with uncertainty, ambiguity, and imprecision in data and decision-making.

In the following we discuss a set of aspects distinctive of fuzzy machine learning:

1. Fuzzy Sets and Membership Functions: In fuzzy machine learning, instead of representing data as crisp values (e.g., 0 or 1), it is represented using fuzzy sets, which allow elements to have degrees of membership between 0 and 1. The membership function defines how much an element belongs to a particular fuzzy set.
2. Fuzzy Inference Systems: Fuzzy inference systems are rule-based systems that use fuzzy logic to make decisions based on input data. These systems consist of fuzzy rules that define relationships between input variables and output variables. Fuzzy inference systems can be used for tasks such as classification, regression, and control.
3. Fuzzy Clustering: Fuzzy clustering algorithms allow data points to belong to multiple clusters with varying degrees of membership. Unlike traditional hard clustering algorithms, where each data point belongs to only one cluster, fuzzy clustering assigns degrees of membership to each data point for every cluster.
4. Fuzzy Decision Trees: Fuzzy decision trees extend traditional decision trees by incorporating fuzzy sets and fuzzy logic into the decision-making process. This allows for more nuanced and flexible decision boundaries.
5. Fuzzy Support Vector Machines (SVM): Fuzzy SVM is an extension of the traditional SVM algorithm that uses fuzzy membership values in the formulation of the decision boundary. It can handle data with uncertainties and overlapping classes more effectively.

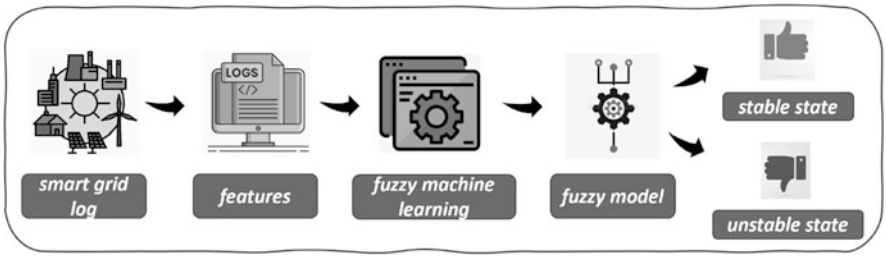


Fig. 1 The workflow of the proposed method for smart grid stability detection by means of fuzzy machine learning

6. Fuzzy Reinforcement Learning: Fuzzy logic can also be integrated into reinforcement learning algorithms to handle the uncertainties and imprecise feedback that often arise in dynamic environments.

Differently to classic machine learning, there are several advantages related to fuzzy machine learning adoption:

1. Robustness to Uncertainty: Fuzzy machine learning methods can handle noisy or imprecise data, making them suitable for real-world datasets with inherent uncertainties.
2. Interpretable Models: Fuzzy machine learning algorithms often provide more interpretable models than their traditional counterparts, making them valuable in decision-making systems where human understanding is essential.
3. Flexibility: Fuzzy machine learning allows for more flexible and adaptive models, particularly useful in situations where the relationships between variables are not well-defined or change over time.

Figure 1 shows the main workflow of the designed method.

The aim of the proposed approach is to perform real-time smart grid stability detection: for this task, it is crucial to timely detect unstable states in a smart grid with the intention of immediately taking action. As shown in Fig. 1 we start from the data collected from the smart grid under analysis (i.e., *smart grid* in Fig. 1).

A smart grid is an interconnected network that facilitates seamless interactions between energy producers and consumers. It primarily consists of two types of nodes: consumption nodes and generation nodes. Generation nodes are responsible for producing energy, such as through wind turbines or photovoltaic panels harnessing solar power. On the other hand, consumption nodes absorb energy to meet the needs of domestic and industrial users, ensuring a reliable supply of renewable energy for their various activities. The smart grid efficiently facilitates the transmission of energy between these nodes, enabling a smooth exchange within the network.

Various smart grid architectures exist, and the method we propose can be applied irrespective of the specific architecture in use. However, for experimentation purposes in this paper, we focus on one of the most commonly used architectures,

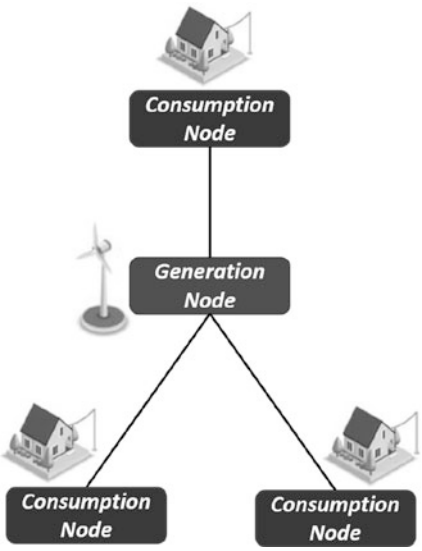


Fig. 2 The four-node star smart grid architecture

known as the four-node star smart grid architecture. In this particular architecture, we have three consumption nodes and one generation node, forming a star schema. Each consumption node is directly connected to the generation node, enabling a direct flow of energy within the network, as shown in Fig. 2.

From the three consumption nodes and the generation node, we gather a set of features aimed to discriminate between smart grid stable and unstable states (i.e., *Features* in Fig. 1).

In particular, 4 features are obtained from each node involved (i.e., network participant) in the smart grid, for a total of 12 features. We consider several feature categories: the first one is the reaction time of each network participant, a real value within the range 0.5–10. From this category, the following features are considered:

- *Feature 0*: it corresponds to the generation node reaction time;
- *Feature 1*: it corresponds to the first consumption node reaction time;
- *Feature 2*: it corresponds to the second consumption node reaction time;
- *Feature 3*: it corresponds to the third consumption node reaction time;

The second feature category is related to the nominal power produced (positive) or consumed (negative) by each network participant, a real value within the range -2.0 to -0.5 for consumers. As the total power consumed is equal to the total power generated, the nominal power supplier nod $= -$ (sum of the nominal power of the three consumption nodes).

From this category, the following features are considered:

- *Feature 4*: it corresponds to the generation node nominal power;
- *Feature 5*: it corresponds to the first consumption node nominal power;
- *Feature 6*: it corresponds to the second consumption node nominal power;

- *Feature 7*: it corresponds to the third consumption node nominal power;

The last feature category is related to the price elasticity coefficient for each network participant, a real value within the range 0.05–1.00;

From this last category, the following features are considered:

- *Feature 8*: it corresponds to the generation node price elasticity coefficient;
- *Feature 9*: it corresponds to the first consumption node price elasticity coefficient;
- *Feature 10*: it corresponds to the second consumption node price elasticity coefficient;
- *Feature 11*: it corresponds to the third consumption node price elasticity coefficient.

We consider the proposed method able to make a prediction in real-time since once the values of the 12 features have been extracted in the same time interval, it is able to make the prediction.

Thus, these features, collected from both stable and unstable smart grid states, represent the input for the fuzzy supervised machine learning algorithms.

A set of feature vectors belonging to stable and unstable smart grid states will be submitted to the *fuzzy model* in order to evaluate the effectiveness of the proposed method to discriminate between *stable* and *unstable* smart grid states in real time.

The evaluation consists of two different stages: (i) we provide a comparison of descriptive statistics of the stable and unstable smart grid state populations and; (ii) classification analysis in order to assess whether the 12 features are able to discriminate between stable and unstable smart grid states.

Regarding descriptive statistics, we present a box plot depicting the distribution of stable and unstable states. The purpose is to illustrate the differences in these distributions and showcase that the features we have considered are promising candidates for effectively discriminating between stable and unstable states.

The classification analysis goal is to verify if the considered features are able to correctly classify between stable and unstable smart grid states. Four fuzzy algorithms of classification were used: NN, FuzzyRoughNN, FuzzyNN, and FURIA. These algorithms were applied to the 12 features (i.e., to the feature vector).

3 The Experiment Analysis

This section is dedicated to the experimental analysis conducted to showcase the efficacy of fuzzy machine learning in detecting stability and instability in smart grids.

We experiment on a dataset composed of smart grid observations obtained from a Kaggle repository.[1] The dataset is freely available for research purposes; it contains

[1] https://www.kaggle.com/code/mineshjethva/power-grid-stability-with-deep-learning

Fig. 3 Box plots related to the F0 (the generation node reaction time) feature

results from simulations of smart grid stability for a four-node star network, equal to the one shown in Fig. 2.

The dataset contains 10,000 distinct observations, but considering that the smart grid we considered in Fig. 2 is symmetric, the dataset can be augmented in 3! times, or 6 times, representing a permutation of the three consumers occupying three consumer nodes. Thus, we consider the augmented version consisting of 60,000 observations, where 38,280 are unstable observations and the remaining 21,720 are stable observations. For each observation, we have the values for the 12 features we described in the previous section with the detail about the smart grid state, i.e., a binary categorical label that can assume the *stable* or the *instable* value. The dataset does not contain missing values.

To perform the experiment, we considered a machine with an i7 8th Generation Intel CPU and 16 GB RAM memory, equipped with Microsoft Windows 10: the model training employed 1 minute and 55.61 seconds on this architecture.

With the aim to understand whether the proposed feature set can be able to discriminate between stable and unstable smart grid states, we consider a way to provide a graphical impact about this, by exploiting box plots. As a matter of fact, box plots are a graphical representation used to visualize the distribution of a dataset and to identify the presence of outliers. They provide a quick and concise summary of the data's central tendency, dispersion, and skewness. Box plots are particularly useful when dealing with large datasets or comparing multiple datasets side by side. Overall, box plots are a valuable tool in exploratory data analysis and are frequently used in various fields, including statistics, data science, and data visualization.

Figures 3, 4, 5, and 6 show the box plots related to the F0 (the generation node reaction time), F5 (the first consumption node nominal power), F8 (the generation node price elasticity coefficient), and F11 (the third consumption node price elasticity coefficient) features relating to the stable and unstable distributions.

Figure 3 is related to the box plots related to the F0 (the generation node reaction time) feature.

From the box plots in Fig. 3, we can note that the two distributions, although partially overlapping, have nonoverlapping areas: for this reason, the F0 (the generation node reaction time) feature can be considered as partially discriminating to discriminate between stable and unstable smart grids.

Figure is related to the box plots related to the F5 (the first consumption node nominal power) feature.

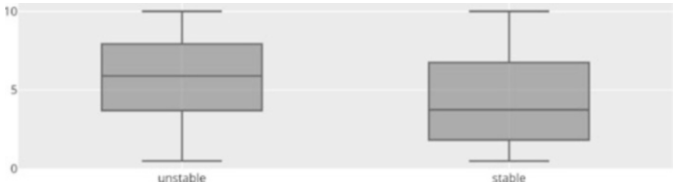


Fig. 4 Box plots related to the F5 (the first consumption node nominal power) feature

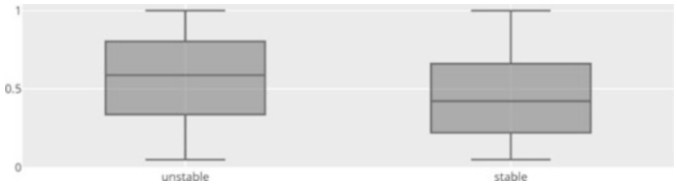


Fig. 5 Box plots related to the F8 (the generation node price elasticity coefficient) feature

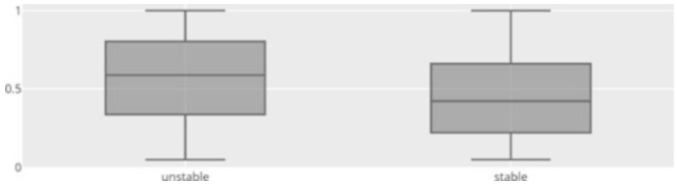


Fig. 6 Box plots related to the F11 (the third consumption node price elasticity coefficient) feature

This feature exhibits behavior very similar to that of the previous feature; for this reason also, this feature can be considered partially discriminating for classifying stable and unstable smart grids.

Figure 5 is related to the box plots related to the F8 (the generation node price elasticity coefficient) feature.

Also for the F8 (the generation node price elasticity coefficient) feature, we note a behavior similar to the two previously analyzed features (i.e., F0 and F5). Also in this case, the unstable features have a higher value than the unstable features.

Figure 6 is related to the box plots related to the F11 feature.

Also in this case we note that the two distributions (stable and unstable) have different instances with different values, even though there are several instances with similar values between stable and unstable instances.

Similar considerations can be made for the other features. From the analysis of the descriptive statistics, it is therefore evident that there is not a predominant feature in discriminatory terms between stable and unstable states, but all the features bring their own contribution: for this reason, the union of different features could lead to the generation of a good classifier able to discriminate between stable and unstable states.

The second step of the experimental analysis is represented by the classification. For training the classifier, we defined *SM* as a set of smart grid logs *(M, l)*, where each M is associated with a label $l \in \{stable, unstable\}$ indicating the smart grid state (i.e., stable or unstable). For each M, we built a feature vector $\mathrm{F} \in R_y$, where y is the number of the features used in training phase ($y = 12$).

For the learning phase, we use a k-fold cross-validation: the dataset is randomly partitioned into k subsets. A single subset is retained as the validation dataset for testing the model, while the remaining $k - 1$ subsets of the original dataset are used as training data. We repeated the process for $k = 10$ times; each one of the k subsets has been used once as the validation dataset. To obtain a single estimate, we computed the average of the k results from the folds.

We evaluated the effectiveness of the classification method with the following procedure:

1. build a training set $T \subset D$;
2. build a testing set $T' = D \div \mathrm{T}$;
3. run the training phase on T;
4. apply the learned classifier to each element of T'.

Each classification was performed using 90% of the dataset as training dataset and 10% as testing dataset employing the full feature set.

The classification analysis is performed using the Weka[2] tool, a suite of machine learning software, employed in data mining for scientific research with the fuzzy algorithms package.[3]

We describe the classification algorithms we employed in order to evaluate the effectiveness of fuzzy machine learning in discriminating between stable and unstable smart grid states:

- *FuzzyNN*: the fuzzy K-nearest neighbors classifier is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression [2]. This algorithm assumes a fixed radius around each element of the dataset, which corresponds to the average distance between this and the rest of the elements of that class;
- *FuzzyRoughNN*: the rationale behind the fuzzy-rough K-nearest neighbors [11] algorithm is that the lower and the upper approximation of a decision class, calculated by means of the nearest neighbors of a test object, provides good clues to predict the membership of the test object to that class;
- *NN*: the fuzzy-rough K-nearest neighbors [11] algorithms is based on the same principle of the previous one;
- *FURIA*: the Fuzzy Unordered Rule Induction Algorithm [10] algorithm extends the RIPPER algorithm [5], a state-of-the-art rule learner. The specificity of this

[2] http://www.cs.waikato.ac.nz/ml/weka/

[3] http://users.aber.ac.uk/rkj/book/wekafull.jar

Table 1 Experimental analysis results

Algorithm	FP rate	Precision	Recall	F-Measure	ROC area	Prediction
NN	0.169	0.91	0.966	0.937	0.977	*Unstable*
	0.034	0.932	0.831	0.879	0.977	*Stable*
	0.12	0.918	0.917	0.916	0.977	*Weighted Avg.*
FuzzyRoughNN	0.18	0.899	0.913	0.906	0.951	*Unstable*
	0.087	0.842	0.82	0.831	0.951	*Stable*
	0.146	0.879	0.879	0.879	0.951	*Weighted Avg.*
FuzzyNN	0.286	0.848	0.905	0.876	0.81	*Unstable*
	0.095	0.81	0.714	0.759	0.81	*Stable*
	0.217	0.834	0.836	0.834	0.81	*Weighted Avg.*
FURIA	0.112	0.935	0.961	0.948	0,966	*Unstable*
	0.039	0.932	0.888	0.910	0.966	*Stable*
	0.085	0.934	0.934	0.933	0.966	*Weighted Avg.*

Table 2 Confusion matrix for the FURIA fuzzy model

a	b	← classified as
3608	145	a = unstable
252	1995	b = stable

algorithm is represented by the fact that it is able to learn fuzzy rules instead of conventional rules and unordered rule sets rather than rule lists. Moreover, to deal with uncovered examples, the algorithm use an efficient rule stretching method. Each individual rule is learned in two steps. The training data, which has not yet been covered by any rule, is thus split into a growing and a pruning set. In the first step, the rule will be specialized by adding antecedents which were learned using the growing set. Afterward, the rule will be generalized by removing antecedents using the pruning set [14].

Table 1 shows the results we obtained from the experimental analysis. The time required for model building is equal to 0.57 seconds for the *NN* and the *FuzzyNN* fuzzy models, while for the *FuzzyRoughNN* model, it is equal to 0.62 seconds. Relating to the *FURIA* model, this last model requires 3878.91 second for model building.

From the results in Table 1, it emerges that the *NN* fuzzy model obtains an average precision and recall equal to 0.91; the *FuzzyRoughNN* model reaches a value equal to 0.87 for average precision and recall, while the *FuzzyNN* model obtains 0.83 for the same metrics. The model obtaining the best performance is the *FURIA* one, with a weighted precision equal to 0.934 and a weighted recall of 0.934.

The confusion matrix for the fuzzy model obtaining the best performances, i.e., *FURIA*, is shown in Table 2.

As shown from the confusion matrix in Table 2 3608, unstable feature vectors were correctly detected in the right category, and 1995 stable feature vectors were correctly in the stable category. Relating to misclassifications, 252 stable feature

Table 3 Rules with a confidence factor greater or equal than 0.98 obtained with the FURIA model

#	Rule	State	CF
1	(F0 in [3.288621, inf]) and (F8 in [0.558881, inf]) and (F1 in [2.907901, inf]) and (F2 in [2.865241, inf]) and		
	(F3 in [2.865241, inf]) and (F9 in [0.434347, inf])	Unstable	1.0
2	(F3 in [3.758101, inf]) and (F11 in [0.552814, inf]) and (F0 in [3.032843, inf]) and (F1 in [2.48691, inf]) and		
	(F2 in [2.48691, inf]) and (F8 in [0.245657, inf])	Unstable	1.0
3	(F1 in [3.746054, inf]) and (F9 in [0.554099, inf]) and (F0 in [2.320936, inf]) and (F2 in [2.48691, inf]) and		
	(F10 in [0.4627, inf]) and (F3 in [2.36366, inf])	Unstable	1.0
4	(F2 in [4.044601, inf]) and (F10 in [0.55003, inf]) and (F0 in [3.461518, inf]) and (F8 in [0.400251, inf]) and		
	(F3 in [2.488351, inf])	Unstable	0.99
5	(F1 in [4.023905, inf]) and (F9 in [0.760109, inf]) and (F0 in [3.461518, inf]) and (F8 in [0.389763, inf])	Unstable	0.99
6	(F2 in [4.039055, inf]) and (F10 in [0.554099, inf]) and (F3 in [3.105594, inf]) and (F11 in [0.406919, inf]) and		
	(F1 in [1.767006, inf]) and (F0 in [1.913214, inf])	Unstable	1.0
7	(F1 in [4.089905, inf]) and (F9 in [0.506574, inf]) and (F3 in [3.34573, inf]) and (F11 in [0.497335, inf]) and		
	(F2 in [2.075289, inf]) and (F8 in [-inf, 0.632616])	Unstable	1.0
8	(F1 in [3.204176, inf]) and (F9 in [0.400352, inf]) and (F2 in [3.577064, inf]) and (F10 in [0.635493, inf]) and		
	(F0 in [2.077598, inf])	Unstable	0.99
9	(F0 in [3.122286, inf]) and (F8 in [0.611277, inf]) and (F1 in [4.06555, inf]) and (F9 in [0.579843, inf])	Unstable	0.98
10	(F2 in [5.685661, inf]) and (F10 in [0.554342, inf]) and (F0 in [4.252039, inf]) and (F8 in [0.323349, inf]) and		
	(F3 in [2.441644, inf]) and (F1 in [1.950486, inf])	Unstable	1.0

vectors were wrongly labeled as unstable, and 145 unstable feature vectors were wrongly detected as unstable.

In Table 3 we show the rules extracted in the *FURIA* model training, with the aim to understand the reason why a certain feature vector is classified as unstable (and thus, providing a kind of global explainability behind the model decision): we can also understand the features that, from the FURIA model point of view, are the most discriminative in order to discern between stable and unstable states.

4 Conclusion and Future Work

To ensure the stability of a smart grid, two essential criteria must be considered. Firstly, the energy generation must constantly match the demand and include a reserve (battery storage) to address immediate outages. Secondly, the smart grid

must maintain sufficient voltage stability throughout its network. In this paper, we proposed a method to detect smart grid stability by leveraging fuzzy supervised machine learning algorithms. The objective is to distinguish between stable and unstable states of the smart grid. We employ four different fuzzy machine learning classification algorithms and achieve excellent precision and recall (both equal to 0.934) when evaluating a dataset comprising 10,000 smart grid observations. A real-world adoption of the proposed consists of the deployment of the proposed model into a server that collects the data from all the nodes and performs the real-time detection of instability situations. In future work, we intend to assess the effectiveness of the proposed method in more complex smart grids with over four nodes and diverse generation nodes. Additionally, we plan to incorporate local explainability alongside global explainability, aiming to provide users with insights into why specific predictions were made. In other words, we seek to grant the method the capability to explain the reasons behind particular outputs based on specific inputs.

Acknowledgments This work has been partially supported by EU DUCA, EU CyberSecPro, EU E-CORRIDOR, PTR 22-24 P2.01 (Cybersecurity), and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the EU – NextGenerationEU projects.

References

1. Aghamohammadi, M.R., Abedi, M.: Dt based intelligent predictor for out of step condition of generator by using PMU data. Int. J. Electr. Power Energy Syst. **99**, 95–106 (2018)
2. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. **6**, 37–66 (1991)
3. Arzamasov, V., Böhm, K., Jochem, P.: Towards concise models of grid stability. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 1–6. IEEE, Piscataway (2018)
4. Breviglieri, P., Erdem, T., Eken, S.: Predicting smart grid stability with optimized deep models. SN Comput. Sci. **2**(2), 1–12 (2021)
5. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123 (1995)
6. Dewangan, F., Biswal, M., Patnaik, B., Hasan, S., Mishra, M.: Smart grid stability prediction using genetic algorithm-based extreme learning machine. In: Electric Power Systems Resiliency, pp. 149–163. Elsevier, Amsterdam (2022)
7. Ezzahra, H.F., Aatila, M., Lachgar, M., Abdali, A.: Predicting smart grid stability using machine learning algorithms. In: 2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC), pp. 1–6. IEEE, Piscataway (2022)
8. Ghosh, A., KOLE, A.: A comparative analysis of enhanced machine learning algorithms for smart grid stability prediction (2021)
9. Gupta, A., Gurrala, G., Sastry, P.: An online power system stability monitoring system using convolutional neural networks. IEEE Trans. Power Syst. **34**(2), 864–872 (2018)
10. Hühn, J., Hüllermeier, E.: Furia: an algorithm for unordered fuzzy rule induction. Data Mining Knowl. Discovery **19**(3), 293–319 (2009)
11. Jensen, R., Cornelis, C.: A new approach to fuzzy-rough nearest neighbour classification. In: International Conference on Rough Sets and Current Trends in Computing, pp. 310–319. Springer, Berlin (2008)

12. Khan, A.A., Wiens, T., Massoth, M.: Stability improvement solution of the smart power grid by an analysis of voltage variation in intelligent buildings. In: SMART 2014: The Third International Conference on Smart Systems, Devices and Technologies, pp. 13–19 (2014)
13. Kreimel, P., Eigner, O., Mercaldo, F., Santone, A., Tavolato, P.: Anomaly detection in substation networks. J. Inform. Secur. Appl. **54**, 102527 (2020)
14. Martinelli, F., Mercaldo, F., Nardone, V., Santone, A.: Car hacking identification through fuzzy logic algorithms. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–7. IEEE, Piscataway (2017)
15. Martinelli, F., Mercaldo, F., Santone, A.: Machine learning for driver detection through can bus. In: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), pp. 1–5. IEEE, Piscataway (2020)
16. Martinelli, F., Mercaldo, F., Santone, A.: A method for intrusion detection in smart grid. Proc. Comput. Sci. **207**, 327–334 (2022)
17. Martinelli, F., Mercaldo, F., Santone, A.: Water meter reading for smart grid monitoring. Sensors **23**(1), 75 (2022)
18. Moldovan, D., Salomie, I.: Detection of sources of instability in smart grids using machine learning techniques. In: 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 175–182. IEEE, Piscataway (2019)
19. Schäfer, B., Grabow, C., Auer, S., Kurths, J., Witthaut, D., Timme, M.: Taming instabilities in power grid networks by decentralized control. Eur. Phys. J. Special Topics **225**(3), 569–582 (2016)
20. Schäfer, B., Matthiae, M., Timme, M., Witthaut, D.: Decentral smart grid control. N. J. Phys. **17**(1), 015002 (2015)
21. Tiwari, S., Jain, A., Yadav, K., Ramadan, R.: Machine learning-based model for prediction of power consumption in smart grid. Int. Arab J. Inf. Technol. **19**(3), 323–329 (2022)
22. Zare, H., Alinejad-Beromi, Y., Yaghobi, H.: Intelligent prediction of out-of-step condition on synchronous generators because of transient instability crisis. Int. Trans. Electr. Energy Syst. **29**(1), e2686 (2019)

On Protection of the Next-Generation Mobile Networks Against Adversarial Examples

Mikhail Zolotukhin, Di Zhang, and Timo Hämäläinen

1 Introduction

Employing artificial intelligence (AI) and machine learning (ML) is forecast to enhance future generation mobile networks by providing means for efficient dynamic resource allocation [8, 19, 46] as well as decreasing transmission overheads [2, 10] and reducing computing costs [61]. Various AI/ML models are therefore proposed to be deployed in 5G for automatic modulation recognition [40], channel estimation in multi-input multi-output (MIMO) frameworks [51], channel decoding [30], energy efficient power allocation [53], network routing optimisation [6], intelligent network slicing [56], and intrusion detection [13]. As dependability of future 5G networks on accurate and timely performance of its AI/ML-driven components is expected to grow, there is an increasing concern that those components can become a high-value attack target since disturbance in their functionality may have a negative impact on the entire network. In particular, due to the open nature of wireless medium, a smart adversary can try to manipulate the inputs of the AI/ML models over the air which may result in significant degradation of the network performance [71, 72].

The study of adversarial machine learning, which focuses on the problem of learning in the presence of adversaries, has recently emerged in many research domains including mobile networking [7, 24, 25]. An adversarial machine learning attack can be carried out during either the training or inference stage. During the training, the adversary aims to poison the training data in such a way that results in the target model making errors later in the inference time [60]. When employing such an attack approach against a 5G network, an adversary can in theory utilise

M. Zolotukhin (✉) · D. Zhang · T. Hämäläinen
Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
e-mail: mizolotu@jyu.fi; zhdizhan@jyu.fi; timoh@jyu.fi

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_11

the end users' equipment to send false signals and messages to the radio access network (RAN) domain after authentication and authorisation in the core network (CN) domain. If a network provider decides to use the data collected to train an AI/ML model, it ends up with the model that is either inaccurate or, in the worst-case scenario, vulnerable to backdoor triggers. Apart from data poisoning via end users' equipment, poisoning attacks may also take place when transmitting model parameters and/or data in certain transmission procedures as RAN can also exchange data with the CN and the operation and maintenance (OAM) domains to collaboratively improve specific end users' quality of experience (QoE) with regard to data rate [54] and to cooperatively enhance the service continuity of end users [33] as well as RAN throughput [37]. Although such threat certainly exists, this attack approach can be successful only in the case of some major flaw in the data processing pipeline, e.g. if the provider decides to use the training dataset obtained from a non-trusted source or the adversary has access to some internal components of the AI/ML model during the training stage which allows it to inject poisonous data.

During the inference, the goal of the adversary is to force the target model to return a certain wrong output by adding an intelligently crafted perturbation to the corresponding input sample [62]. This category of attacks, which is often referred to as adversarial examples, is a more realistic threat to mobile networks that employ AI/ML in their intelligent components as it does not require access to the dataset and the target model itself during the training. The only requirement is that the model is supposed to have some sort of interface which can be used by the adversary over the air to manipulate inputs of the model [24, 72]. Such attacks based on adversarial example generation at the inference stage are usually classified into either white-box or black-box category depending on the information available to the adversary. The former includes cases when the adversary has perfect knowledge of either the machine learning model or the data used for its training or both of them. In the latter scenario, the adversary's only capability is to observe labels assigned by the model for the inputs supplied. Attacks from the black-box category are more practical for real-world adversaries with knowledge about neither the model nor the training data and therefore in our research we focus on this type of attack.

Our contribution to the research domain of adversarial machine learning in the context of mobile networking is twofold. We first identify potential attack vectors against multiple applications of AI/ML in 5G wireless networks by evaluating various black-box adversarial example generation algorithms. We then implement and compare several defence methods that can be used by service providers to protect their mobile networks from these attacks. The rest of the document is organised as follows. Adversarial example generation techniques in the RAN domain as well as several defence strategies are briefly discussed in Sect. 2. Section 3 describes the use case scenarios selected for evaluation in this study. Numerical simulation results for these use cases are presented in Sect. 4. Section 5 concludes the paper and outlines future work.

2 Theoretical Background

In this section, we first summarise the most popular algorithms for adversarial example generation and discuss their transferability to the RAN domain. After that, we overview various defence strategies such as adversarial example detection and rejection as well as model hardening techniques.

2.1 Adversarial Examples in Mobile Networks

The vast majority of AI/ML applications proposed to be deployed in the next-generation mobile networks rely on deep learning which has gained increasing popularity in recent years due to its supremacy in terms of accuracy when trained with huge amounts of data [20]. A deep neural network most of the time has a differentiable loss function and uses a gradient-based optimiser during the training. This enables gradient-based adversarial example generation by modifying an input sample in the direction of the gradient of the loss function with respect to the input sample [15, 28, 41, 62]. The size of the step taken towards this direction is usually limited by the perturbation budget, i.e. the size of the perturbation allowed in the research problem given. Several adversarial perturbation crafting algorithms rely on computing the perturbation iteratively with a step size smaller than the perturbation size allowed [28]. In this case, the intermediate perturbation values are clipped after each step to ensure that they satisfy the constraints imposed on the adversarial perturbation budget and the final perturbation is calculated by adding these intermediate values together [31].

The approach described above is white-box as the adversary would require full access to the model parameters and the loss function to calculate the gradient for the input sample. In the black-box settings, the target model can remain unknown to the adversary. However, it is usually assumed that the adversary has an ability to query an input sample to the target classifier in order to adjust the perturbation value. The first fundamental study in this research area [43] proposes a transfer based attack approach which relies on information about training samples without the knowledge of the training model. The adversary's strategy is to learn a substitute for the target model using a synthetic dataset carefully generated by the adversary and labelled by observing the target model output. Once the substitute model has been trained, a white-box attack is used to synthesise perturbations for this substitute. The adversary expects the target model to misclassify the resulting perturbed samples due to transferability between deep learning models. The approach can therefore be employed against various neural networks with different rates of success. Several studies propose to search for the perturbation without training a substitute model [5, 9]. These black-box attacks are based on estimating gradient direction at the decision boundary based solely on access to the target model decisions for the inputs queried.

Speaking of adversarial examples in the mobile networking domain, there are multiple use case scenarios studied in recent scientific literature. Probably, the most well-studied use case involves compromising the integrity of the modulation recognition classifiers [22–24, 65]. Automatic modulation recognition can be implemented in adaptive transceivers to automatically switch modulations based on the channel conditions without the need for a feedback channel between the transmitter and the receiver. AI/ML models have been recently proposed to be deployed for this purpose as they are able to classify the modulation type based on the received signal strength (RSS) values [35, 40, 45]. In particular, this would allow a user equipment (UE) to recognise the modulation type used and initiate the contention-based random access procedure to attach to the network. Theoretically, an adversary can use the broadcast nature of the wireless channel and perturb the input data, i.e. the series of RSS values, by introducing a noise in such a way that would lead to the modulation scheme being misclassified by the AI/ML model which would lead to the network access procedure failure.

Another category of the adversarial example attacks in mobile networks includes attacks targeting AI/ML models for intelligent mmWave beam selection. For example, in the 5G new radio (5G-NR) initial access (IA) procedure, a gNodeB (gNB) periodically broadcasts synchronisation signal blocks (SSBs) that correspond to different beam(s). An UE may select the most suitable beam based on the output of the AI/ML model trained to select the beam based only on a subset of the received SSB measurements in order to reduce the transmission overhead [10]. The UE then notifies the gNB about the SSB selected with an msg1 message. Both SSB(s) and msg1 may be transmitted without security protection. An adversary can therefore use this information to generate such a signal which causes the UE to select a wrong beam and fail to access the network [25].

Other less studied adversarial generation examples in the domain of mobile networking include attacks against AI/ML-based signal decoding [50, 65], channel estimation in MIMO [29], power allocation frameworks [26, 32, 66], an attack against an intelligent scheduling framework for the spectrum sharing in 5G with citizens broadband radio service [52], and attacks against the models which are trained to allocate physical resources to the network slices [55, 57].

2.2 Defence Strategies

From the methodology point of view, adversarial example generation attacks described above resemble radio jamming, i.e. an adversary most of the time uses a malicious fake BS (FBS) to emit a radio signal on top of existing transmissions over the air to change the model input data in such a way that the target AI/ML model's output is incorrect which in turn may negatively affect the functionality of the corresponding network component in which the model is deployed. Depending on the network component under attack, the impact may vary. In the majority of cases, the effect is similar to radio jamming and denial-of-service attacks, i.e. deterioration

of the network data rate which can lead to mobile customers being unable to use network services [17]. For this reason, mitigation measurements from RAN-3 of the Baseline Security Controls can be employed [16]. These include but are not limited to monitoring for and responding to traffic fluctuations, unusual handover patterns, dead spots, and service disruptions.

On the AI/ML model defence side, two different strategies are usually considered: runtime detection of adversarial inputs and model hardening. The former is usually designed to work under the so-called manifold hypothesis which assumes that normal data samples lie in a low-dimensional manifold embedded in a high-dimensional space [12]. Such manifold-based defences work by identifying adversarial points from their distance to the manifold [59]. In practice, such detection can be implemented by training a two-class detector network which obtains inputs from intermediate feature representations of the target classifier and is trained to discriminate between samples from the original dataset and adversarial examples in a supervised way [36]. Alternatively, an anomaly detection approach can be employed in order to train the model of normal behaviour and classify the samples that deviate significantly from the norms established as malicious ones [67]. Such an approach will in theory generate more false alarms than the supervised one, since not every anomaly corresponds to an adversarial example. However, it is worth noticing that since adversarial ML has recently attracted significant attention, novel attack approaches are constantly emerging. At the same time, a supervised detection model trained using adversarial examples generated with known attack algorithms may not generalise well for adversarial examples generated by novel ones that have not been present in the training dataset which may result in lower detection rates.

Among the model hardening methods, a widely explored approach is to augment the training data of the AI/ML model with adversarial examples [15]. Another approach is input data preprocessing, often using non-differentiable or randomised transformation [18], transformations reducing the dimensionality of the inputs [69], or transformations aiming to project inputs onto the normal data manifold [34]. Other model hardening approaches involve special types of regularisation during model training [48] or modifying elements of the classifier's architecture [70].

3 Use Cases

First, we briefly summarise three use cases, attacks against which are evaluated in this study. The use case scenarios include modulation recognition [40], optimal beam selection based on RSS measurements for a subset of beams [10], and jamming detection [44]. We focus on these particular use cases for three main reasons: first, there is a clear benefit of AI/ML deployment; second, the corresponding study provides detailed information on how to implement and train the AI/ML model proposed; third, there is a room for an adversarial example generation attack against the resulting AI/ML framework. Below is the brief summary of the use cases

mentioned followed by a detailed description of the datasets and models used in each use case.

1. *Modulation recognition.* Study [40] proposes a deep neural network enabled modulation recognition approach based on features extracted from complex baseband time series representations of the received signals at various signal-to-noise ratio (SNR) levels. Time series of the received signal acts as the input, the output is the modulation type.
2. *Beam selection.* In [10], a deep learning solution for fast and accurate IA is proposed. The target model is trained by feeding the RSS values from a subset of beams as the input. The output of the network consists of the probabilities of being the optimal vector calculated for each beam in the whole set.
3. *Jamming detection.* Study [44] focuses on deploying an AI/ML-based detection of jamming attacks on unmanned aerial vehicles (UAVs) that operate using orthogonal frequency-division multiplexing (OFDM) communication. In particular, authors attempt to detect and classify multiple jamming attack types which include barrage, single tone, successive pulse (SP), and P-aware (PA). Input features include average received signal and noise power.

3.1 Data

In the modulation recognition use case, RadioML dataset [39] is employed for training AI/ML models. To generate this dataset, real voice and text datasets are modulated onto the signal using a block randomiser to whiten the data in the case of digital modulation to ensure bits are equiprobable [39]. To simulate radio channel effects, various robust models are employed including time-varying multipath fading of the channel impulse response, random walk drifting of carrier frequency oscillator and sample time clocks, and additive Gaussian white noise. The resulting synthetic signal sets are then passed through harsh channel models which introduce unknown scale, translation, dilation, and impulsive noise onto our signal. Each of the resulting 128-sample time series is then scaled to unit energy. The version of the dataset we are using in our experiments (RadioML 2016.10A) consists of 11 modulations: 8 digital and 3 analog modulations, all are widely used in wireless communications systems all around the world. These include BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, and PAM4 for digital modulation and WB-FM, AM-SSB, and AM-DSB for analog modulation. The data is modulated at a rate of roughly eight samples per symbol with a normalised average transmit power of 0 dB.

In the beam selection use case, DeepMIMO dataset [1] is used. This dataset is essentially a light-weight massive MIMO mmWave simulator. Given a scenario and input parameters, it generates channel matrices for each gNB-UE ray-tracing path. In our use case, outdoor scenario O1 is used. One base station (BS 3) is selected as the serving BS and users are assumed to be located between rows 700 and 1300.

The operating frequency is 28 GHz, whereas the gNB and UE antenna shapes are equal to $(1, 64, 1)$ and $(1, 1, 1)$, respectively. Other input parameters can be found in [3]. In order to form the input data, a subset of channel values for randomly picked 25% of the beams is selected. This subset remains the same for all the data samples during the training, validation, and inference stage. The aim is to select such a beam that maximises the theoretical achievable data rate which can be calculated as follows: $r = \log_2(1 + H^{*T} f)$, where H is the channel matrix and f is a column vector from codebook F. For each data point, the mmWave beam in the codebook which provides the highest data rate for the channel matrix given is calculated. The corresponding one-hot vector that indicates the index of this optimal beam acts as the output data point. It is worth mentioning we use a simple quantised beam steering codebook where the i-th beam for $i = 1, 2, \ldots, |F|$ is defined as $f_i = a(\frac{2\pi i}{|F|})$, with a representing the mmWave array response vector [3].

Finally, in the case of the jamming detection problem, we use the dataset obtained with real equipment by authors of study [44]. Input features in the dataset provided include subcarrier spacing, symbol time, subcarrier length, cyclic prefix length, average received power, threshold, average signal power, average noise power, and SNR. In our experiments, we only use the features that can be affected by an adversary over the air, i.e. average received power, average signal power, average noise power, and SNR. Each sample in the dataset is labelled as either normal or as the one corresponding to a jamming attack.

3.2 Models

The models employed in study [40] for automatic modulation recognition are a fully connected neural network and two convolutional neural networks (CNNs). The model that provides the best results in terms of the prediction accuracy is the CNN with 2 convolutional layers of 256 and 80 filters followed by 1 fully connected layer of 256 neurons. In the original study on intelligent beam selection [10], the AI/ML model is a fully connected neural network with 5 hidden layers of 32, 64, 128, 64, and 32 neurons, respectively. Finally, in the jamming detection use case, one of the models tested is a deep neural network; however, the authors do not provide any information on the model architecture [44]. For this reason, we have tested several neural network models and found out that a simple fully connected neural network with 2 layers of 1024 neurons is able to classify samples that correspond to jamming attacks with 100% accuracy.

Unfortunately, when tested using the datasets generated, the modulation recognition and the beam selection models from [40] and [10], respectively, result in extremely low classification accuracy values. For this reason, in our experiments, we build another model in each of these two use cases. As one can notice, the inputs to the target beam selection model can be interpreted as images with height and width being equal to the number of antennas and the number of OFDM subcarriers, whereas the real and imaginary part of the signal can be interpreted as the colour

channels [68]. CNNs are usually employed in image related problems as they allow for automatic extraction of low-level features such as edges, colour, gradient orientation, and several others. Without loss of generality, we rely on CNN in the modulation recognition use case with the real and imaginary part of the signal as previously playing the role of the colour channels similarly to how it has been done in study [40]. However, since data samples in the modulation recognition use case are one-dimensional sequences of symbols, strides of size one with the same padding are used in convolution layers.

In order to find an optimal CNN architecture for the beam selection and modulation recognition use cases, automated machine learning (AutoML) approach can be employed. AutoML is usually defined as producing maximum performance from learning tools without human assistance. In our experiments, we use Auto-Keras [21] which is an efficient open-source neural architecture search system which relies on Bayesian optimisation [58]. In order to generate CNN architectures, we use an image input block followed by a normalisation and an image augmentation block. After that, convolutional blocks, each of which consists of several convolutional, dense, max-pooling, and dropout layers, are used to extract the necessary features. The architecture search is carried out for 100 trials. The resulting neural network architectures later used in the experiments in this study can be found in Table 1.

To train the models generated, each of the datasets obtained is divided into three parts: training (50%), validation (30%), and inference (20%). The training parts are used to train the corresponding AI/ML models, whereas the main function of the validation parts is to control the models' overfitting. The inference parts are then used to evaluate the models. Early stopping is employed in order to stop the training when the validation loss starts increasing. Speaking of the loss function, standard categorical cross-entropy is used for all the classification models trained.

Table 1 AI/ML model architectures and metric values

Use case	Modulation recognition	Beam selection	Jamming detection
Layers	Input(128, 1, 2)	Input(16, 32, 2)	Input(4)
	Normalisation()	Normalisation()	Normalisation()
	Conv2D([7, 7], 512, ReLU)	Conv2D([3, 3], 32, ReLU)	Dense(1024, ReLU)
	Conv2D([7, 7], 32, ReLU)	Conv2D([3, 3], 64, ReLU)	Dropout(0.5)
	Dropout(0.25)	MaxPooling([2, 2])	Dense(1024, ReLU)
	Conv2D([7, 7], 64, ReLU)	Conv2D([3, 3], 512, ReLU)	Dropout(0.5)
	Conv2D([7, 7], 32, ReLU)	Conv2D([3, 3], 256, ReLU)	Dense(2, Softmax)
	Dropout(0.25)	MaxPooling([2, 2])	
	Flatten()	GlobalAveragePooling()	
	Dropout(0.5)	Dropout(0.5)	
	Dense(11, Softmax)	Dense(64, Softmax)	
Parameters	1,099,403	1,510,880	1,056,770
Loss	1.094901	0.063998	5.446e–07
Accuracy	61.64%	98.60%	100.00%

The training is carried out in batches of 512 with learning rate equal to 0.0025. The classification accuracy values obtained when applying the models trained to the inference parts of the corresponding datasets can also be found in Table 1. Despite the relatively low accuracy value obtained in the modulation recognition use case, this result is somewhat in line with the original study, according to which AI/ML-based modulation recognition models allow for accurate modulation recognition only in the case of high SNR values [40].

4 Numerical Simulations

In this section, we first evaluate various black-box adversarial example attacks for the use cases described in Sect. 3. After that, we implement and test several anomaly detection algorithms that can be used for adversarial example detection. Finally, we apply a few model hardening techniques and evaluate their impact on the attack efficiency.

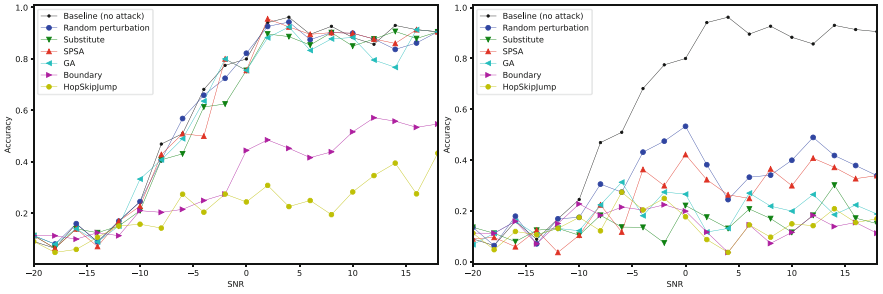
4.1 Attack Algorithm Evaluation

In our experiments, we first employ the attack approach which relies on the transferability property of deep neural networks to generate adversarial examples as it has been proposed in study [43]. For this attack, the neural networks found in the corresponding original studies [10, 40] and [44] are used as the substitute models which are trained using 10% of the samples from the original datasets. To attack these substitute models, we use the white-box algorithm called projected gradient descent (PGD). This method relies on the principle of modifying an input sample in the direction of the gradient of the loss function as it has been briefly described in Sect. 2. In addition, we test the following black-box attacks that do not require training a substitute model: simultaneous perturbation stochastic approximation (SPSA) [64], the attack based on genetic algorithm (GA) [4], boundary attack [5], and HopSkipJump [9]. Implementations of the aforementioned white-box and black-box attacks for the experiments are taken from the following adversarial example generation frameworks: Cleverhans [42], Adversarial Robustness Toolbox (ART) [38], and Foolbox [47]. All the aforementioned algorithms are executed with two different perturbation budget sizes that are relative to the original input and equal to $0.1||X||$ and $1.0||X||$, where $||X||$ is the average second norm for the samples of the corresponding dataset.

The attack efficiency is evaluated based on the effect it produces upon the component that uses the AI/ML model under attack. In the modulation recognition and jamming detection use cases, the attack efficiency is evaluated by comparing the prediction accuracy before and after the attack has been carried out. The less accurate the target model predictions once the attack has been conducted, the

Table 2 The detrimental effect of the adversarial perturbation on the performance of three network components analysed for different attack algorithms and perturbation budgets given in per cent of the input. The metrics are respectively modulation recognition accuracy, theoretical achievable data rate, and jamming detection accuracy. Each metric value presented is relative to the baseline, i.e. when there is no attack taking place. The less the value, the more efficient the attack

	Use case and perturbation budget (in per cent of the input)					
	Modulation recognition		Beam selection		Jamming detection	
Attack algorithm	10%	100%	10%	100%	10%	100%
Random	0.985531	0.495177	0.999846	0.986819	0.896484	0.546875
Substitute	0.946945	0.249196	0.995224	0.905000	0.405273	0.430664
SPSA	0.974277	0.395498	0.999836	0.975431	0.841797	0.531250
GA	0.963023	0.303859	0.999609	0.975191	0.557617	0.557617
Boundary	0.551447	0.239550	0.972949	0.301807	0.633789	0.351562
HopSkipJump	**0.366559**	**0.234727**	**0.888945**	**0.195502**	**0.403320**	**0.000000**

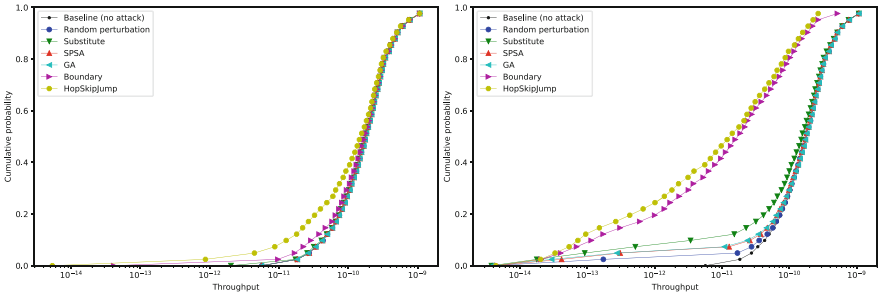


(a) 10% of the input. (b) 100% of the input.

Fig. 1 Dependence of the modulation recognition accuracy on the SNR level when employing various black-box attack algorithms with different adversarial perturbation budget limit values

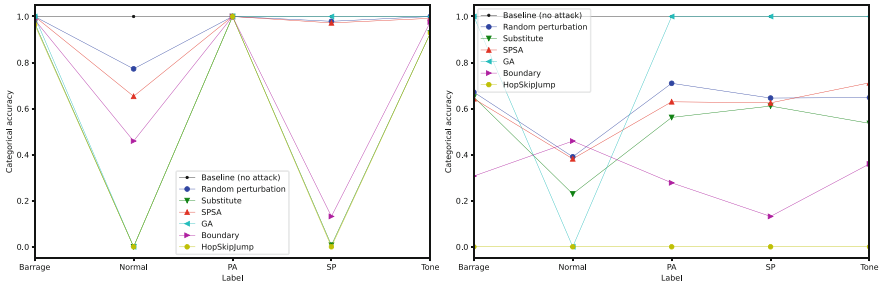
more efficient the attack algorithm. In the beam selection use case, the evaluation metric is the theoretical achievable data sum rate that can be calculated as shown in Sect. 3. All the black-box attack evaluation results for the use cases selected can be respectively found in Table 2 and Figs. 1, 2 and 3. In addition to the perturbations generated with the algorithms mentioned, we also show the effect of a randomly generated perturbation for each use case. In the table, values of the metric selected for each use case are compared to each other for different attacks and adversarial perturbation budget values. All the metric values presented are calculated as percentages of the baseline metric values when there is no attack.

As one can notice when looking at the results presented, adversarial example attack impact on the target model performance is much more significant compared to the random perturbation. Speaking of the attack algorithm comparison results, HopSkipJump algorithm provides the best results in all of the use cases evaluated. This is an iterative algorithm; it starts from a point that is already adversarial, and then at each iteration the following three steps are carried out: estimation of the



(a) 10% of the input. (b) 100% of the input.

Fig. 2 Dependence of the sum-rate CDF on the beam selected based on the beam subset when employing various black-box attack algorithms with different adversarial perturbation budget limit values



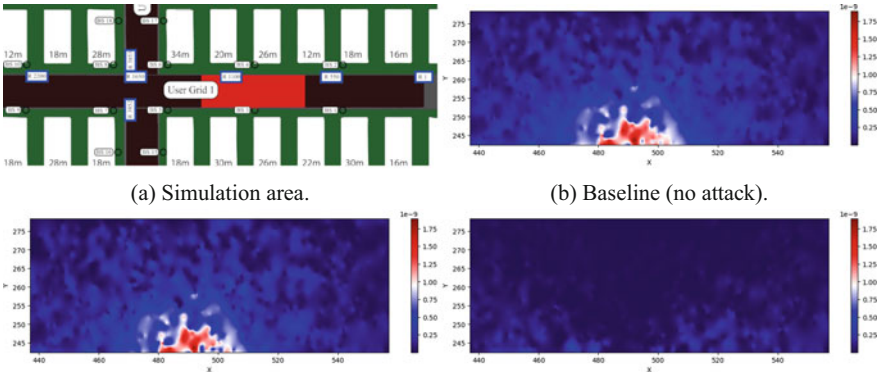
(a) 10% of the input. (b) 100% of the input.

Fig. 3 Dependence of the jamming detection accuracy on the signal type when employing various black-box attack algorithms with different adversarial perturbation budget limit values

gradient direction, step-size search via geometric progression, and boundary search via a binary search [9]. This algorithm is computationally expensive and takes a long time to execute. For this reason, during experiments we had to adjust its default parameters in order to obtain results in a reasonable amount of time.

In the modulation recognition use case, the boundary attack algorithm also shows promising results. This algorithm is essentially simplified version of HopSkipJump as it starts from a point that is already adversarial and then performs a random walk along the boundary between the adversarial and the non-adversarial region such that it stays in the former while the distance towards the target legitimate point is reduced [5]. Both the length of the total perturbation of the adversarial sample and the length of the step towards the original input are adjusted dynamically similarly to a trust region method. When the perturbation is comparable in size with the original input, the negative impact on the target metric is significant for each attack algorithm: the classification accuracy is reduced by more than two times from its original value.

In the beam selection use case, the situation is more optimistic for the mobile network operator in the case of lower adversarial perturbation size values as the data



(a) Simulation area. (b) Baseline (no attack).

(c) HopSkipJump attack (10% of the input). (d) HopSkipJump attack (100% of the input).

Fig. 4 Dependence of the theoretical data rate on UE location detection in the beam selection use case. Figure (**a**) shows the simulation area of scenario O1 from DeepMIMO dataset (UE locations under attack are highlighted with red). Theoretical achievable data rates of UEs located in the simulation area in the case there is no attack are shown in Figure (**b**). Figures (**c**) and (**d**) demonstrate the negative effect of the HopSkipJump attack on the data rate for two different perturbation budget values equal respectively to 10% and 100% of original RSS

rate is reduced by only about 11% in the worst-case scenario. In the case of higher perturbation budget values, as previously, boundary and HopSkipJump algorithms provide the best results in terms of network throughput reduction. To visualise the attack effect, we plot the data rate values obtained during the boundary attack for different UEs located in the simulation area. As one can see from Fig. 4, in the case of lower perturbation budget values the attack targets the UEs located on the edge of the cell as output labels of the corresponding data samples are closer to the decision boundary and therefore the probability of these samples being misclassified by the target model is higher [41]. When the adversarial perturbation is comparable in size with the original input signal, the attack effect is devastating as the network throughput is reduced significantly for most of the UEs in the cell.

Finally, in the jamming detection use case, HopSkipJump attack algorithm allows the adversary to achieve 100% accuracy reduction in the case of higher perturbation budget values. In addition, the attack based on GA may result in quite a significant negative effect in this case. This attack aims to maximise the fitness function designed in such a way that it allows the adversary to increase the probability of the target class and at the same time decrease the probability of all other classes. In our experiments, the attack is non-targeted, i.e. we assume the adversary aims to cause the target classifier to return any incorrect label. Therefore, in this case, the target label can be any but the original one. Figure 3b reveals that the algorithm targets normal data samples rather than the ones corresponding to the jamming attack, which will result in the increased number of false alarms. In the case of lower perturbation budget values, HopSkipJump and substitute attack algorithms provide for the best results in terms of detection accuracy reduction. In addition to increased

numbers of false alarms, these algorithms cause the data samples that correspond to successive-pulse jamming being misclassified as normal ones by the target AI/ML model as it can be seen from Fig. 3a.

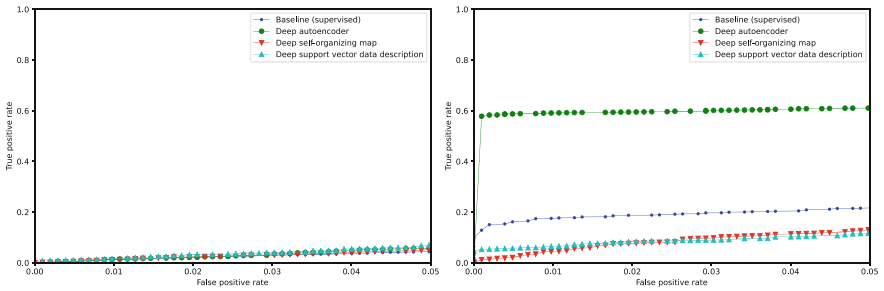
4.2 Attack Detection

As mentioned in Sect. 2, the most straightforward approach to detect adversarial examples is to connect a classification head to the target model and train the resulting classifier to distinguish normal samples from the ones corresponding to the attack [36]. We evaluate such supervised approach and compare it against the one based on anomaly detection using the following three deep anomaly detection algorithms: deep autoencoder (AE), deep self-organising map (SOM) [14], and deep support vector data description (SVDD) [49]. The deep autoencoder is a neural network model which aims to adjust its trainable parameters in such a way that the output layer is equal to the input one despite the information bottleneck caused by the hidden layers. The role of the loss function is often played by the reconstruction error which is the difference between the input and the output. An anomalous data point fed to the autoencoder model often results in something that is quite different from the expected output, and therefore a large error value. The deep self-organising map consists of an autoencoder and a self-organising map which is essentially a grid of neurons, each of which is fully connected to the previous layer of the model which in the case of deep SOM is the last encoding layer of the autoencoder. Each neuron of the SOM is associated with a prototype vector, the dimension of which is equal to the dimension of its input vectors. At each training step, one feature vector from the input dataset is picked up randomly, and the distance between it and all the prototype vectors is calculated. The prototype vector that is the least distant from input is denoted as its best matching unit (BMU). The input vector is mapped to the location of the best matching unit, and the prototype vectors of the SOM are updated so that the vector of the BMU and its topological neighbours are moved closer to the input sample [27]. The loss function of the resulting model is the weighted sum of the autoencoder's reconstruction error and the distance to the SOM's best matching unit. Finally, the deep support vector data description is the neural network model that learns a transformation from an input space to an output space which attempts to map most of the data network representations into a hypersphere of minimum volume. During the inference stage, mappings of normal examples fall within, whereas mappings of anomalies fall outside the hypersphere.

To train each of the detection models mentioned above, we use samples from the original datasets. For unsupervised anomaly detection methods, having only normal non-perturbed samples is enough to carry out the training. In the case of the supervised model, we augment the original datasets with adversarial examples generated with the following white-box attack algorithms: one-shot fast gradient sign method (FGSM) [15], basic iterative method (BIM) [28], momentum iterative method (MIM) [11], and already mentioned above projected gradient descent (PGD)

Table 3 The adversarial example detection accuracy calculated for different supervised and unsupervised algorithms and perturbation budgets given in per cent of the input. The greater the value, the more efficient the detection algorithm

	Use case and perturbation budget (in per cent of the input)					
	Modulation recognition		Beam selection		Jamming detection	
Detection model	10%	100%	10%	100%	10%	100%
Supervised	0.501855	0.584473	**0.770508**	**0.922168**	0.934766	0.963184
Deep AE	0.504980	**0.791602**	0.500293	0.505176	0.860449	0.901953
Deep SOM	0.502539	0.540039	0.524512	0.774219	0.862988	0.891797
Deep SVDD	**0.510449**	0.533594	0.525293	0.660840	**0.994824**	**0.999902**

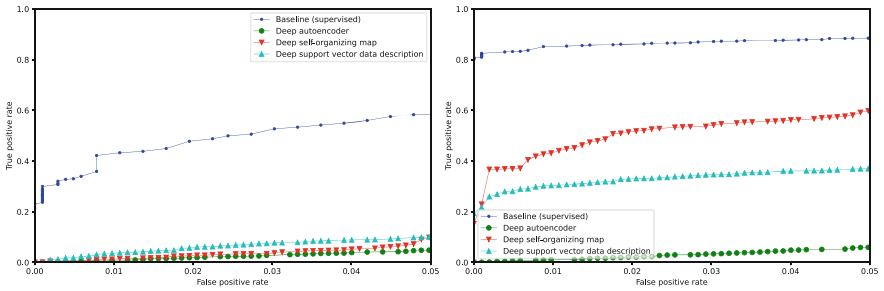


(a) 10% of the input. (b) 100% of the input.

Fig. 5 Dependence of the true positive rate on the false positive rate when detecting black-box adversarial examples in the modulation recognition use case with two different adversarial perturbation budget limit values with the help of baseline supervised and various unsupervised anomaly detection methods

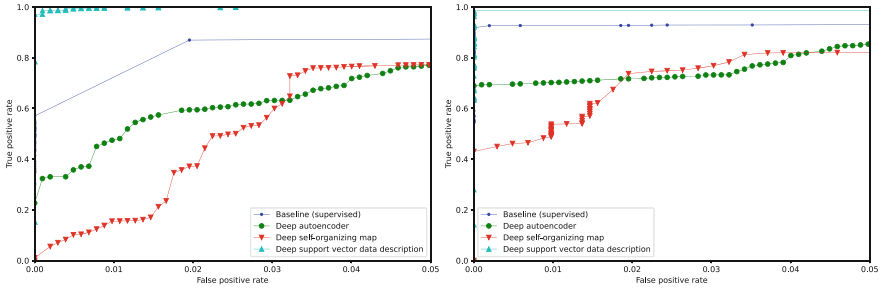
[31]. As previously, dropout and early stopping are used to combat overfitting. Training is carried out in batches of 512 with learning rate 0.0025. The results, namely, anomaly detection accuracy as well as the dependence of the true positive rate (TPR) on the false positive rate (FPR) for each of the algorithms tested, can be found in Table 3 and Figs. 5, 6 and 7. It is worth noticing that we are only interested in the detection rates obtained for low FPR numbers, in particular lower than 5%.

As one can see from the results obtained, in two of the three use cases studied, adversarial examples are almost impossible to detect in the case of lower perturbation budget values. Only in the jamming detection use case, a significant portion of the malicious samples is classified accurately. In other two use cases, when the perturbation budget is equal to the input size, the percentage of adversarial examples that can be identified under the constraint of keeping the numbers of false alarms low ranges from 60 to 80%. Another interesting observation is that some of the unsupervised anomaly detection methods tested provide slightly better detection rates compared to the supervised one. For example, in the modulation use case the deep autoencoders outperform other algorithms in terms of detection accuracy, whereas the deep SVDD provides the most promising results for detecting adversarial examples generated against the jamming detection model. The relatively



(a) 10% of the input. (b) 100% of the input.

Fig. 6 Dependence of the true positive rate on the false positive rate when detecting black-box adversarial examples in the beam selection use case with two different adversarial perturbation budget limit values with the help of baseline supervised and various unsupervised anomaly detection methods



(a) 10% of the input. (b) 100% of the input.

Fig. 7 Dependence of the true positive rate on the false positive rate when detecting black-box adversarial examples in the jamming detection use case with two different adversarial perturbation budget limit values with the help of baseline supervised and various unsupervised anomaly detection methods

poor performance of the detectors that follow the supervised approach in those cases can be explained by the fact that the adversarial examples generated with black-box algorithms under consideration have not been present in the corresponding training datasets and supervised models in general are not designed to deal with the samples from unknown classes.

4.3 Model Hardening

Concerning the model hardening approach for AI/ML model protection against adversarial examples, we test the two following popular techniques: adversarial training [15] and label smoothing [63]. The former allows one to inject adversarial

Table 4 The detrimental effect of the adversarial perturbation on the modulation recognition accuracy for different attack algorithms and perturbation budgets given in per cent of the input in the case of several model hardening techniques being applied. Each metric value presented is relative to the baseline, i.e. when there is no attack taking place. The less the value, the more efficient the attack

	Model hardening method and perturbation budget (in per cent of the input)					
	Baseline (no defence)		Adversarial learning		Label smoothing	
Attack algorithm	10%	100%	10%	100%	10%	100%
Random	0.985531	0.495177	0.978784	0.479491	0.996825	0.457143
Substitute	0.946945	0.249196	0.851485	0.422914	0.839683	**0.146032**
SPSA	0.974277	0.395498	0.957567	0.340877	0.979365	0.428571
GA	0.963023	0.303859	0.975955	0.275813	0.971429	0.261905
Boundary	0.551447	0.239550	0.527581	0.155587	0.544444	0.271429
HopSkipJump	**0.366559**	**0.234727**	**0.364922**	**0.147100**	**0.376190**	0.268254

Table 5 The detrimental effect of the adversarial perturbation on the data rate in the beam selection use case for different attack algorithms and perturbation budgets given in per cent of the input in the case of several model hardening techniques being applied. Each metric value presented is relative to the baseline, i.e. when there is no attack taking place. The less the value, the more efficient the attack

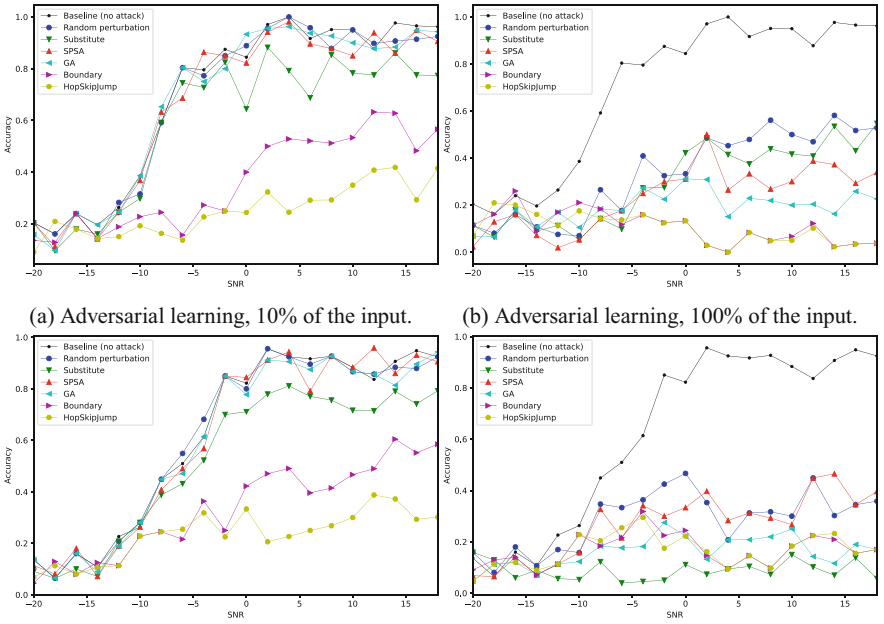
	Model hardening method and perturbation budget (in per cent of the input)					
	Baseline (no defence)		Adversarial learning		Label smoothing	
Attack algorithm	10%	100%	10%	100%	10%	100%
Random	0.999846	0.986819	0.999998	0.988811	0.999698	0.987671
Substitute	0.995224	0.905000	0.998609	0.957780	0.988861	0.803566
SPSA	0.999836	0.975431	0.999960	0.930313	1.000051	0.976413
GA	0.999609	0.975191	0.999952	0.932794	0.999600	0.971647
Boundary	0.972949	0.301807	0.983622	0.368111	0.976489	0.320126
HopSkipJump	**0.888945**	**0.195502**	**0.890961**	**0.208198**	**0.896061**	**0.199990**

examples during training to improve the generalisation of the target model and in theory make it more resilient to adversarial samples, whereas the latter relies on mixing the original ground-truth label distribution with some fixed distribution aiming to make the target model less confident over its predictions. In our experiments, we generate samples for adversarial training using the same white-box algorithms as we have used when evaluating detection methods, i.e. FGSM, PGD, BIM, and MIM. For label smoothing, we use the uniform distribution with the weight of this distribution being equal to 0.1 divided by the number of classes similarly to how it has been done in the original study [63].

The results are presented in Tables 4, 5 and 6 and Figs. 8, 9 and 10. As one can immediately notice, there are no significant model performance improvements when employing the aforementioned model hardening techniques in each of the use case scenarios studied. In the modulation recognition use case, some significant accuracy increase can be observed when employing adversarial training against the attack that relies on using a substitute model. On the other hand, label smoothing

Table 6 The detrimental effect of the adversarial perturbation on the jamming detection accuracy for different attack algorithms and perturbation budgets given in per cent of the input in the case of several model hardening techniques being applied. Each metric value presented is relative to the baseline, i.e. when there is no attack taking place. The less the value, the more efficient the attack

	Model hardening method and perturbation budget (in per cent of the input)					
	Baseline (no defence)		Adversarial learning		Label smoothing	
Attack algorithm	10%	100%	10%	100%	10%	100%
Random	0.896484	0.546875	0.884766	0.709961	0.889648	0.574219
Substitute	0.405273	0.430664	**0.438477**	0.422852	0.419922	**0.000000**
SPSA	0.841797	0.531250	0.836914	0.781250	0.844727	0.583008
GA	0.557617	0.557617	0.945312	0.557617	0.560547	0.557617
Boundary	0.633789	0.351562	0.780273	0.527344	0.625977	0.286133
HopSkipJump	**0.403320**	**0.000000**	0.500000	**0.186523**	**0.416992**	0.002930



(a) Adversarial learning, 10% of the input.

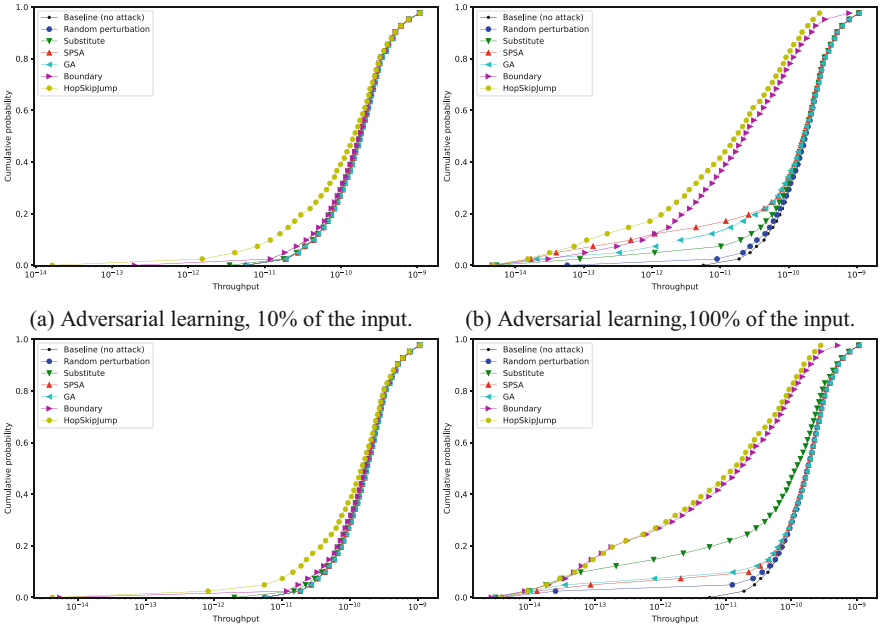
(b) Adversarial learning, 100% of the input.

(c) Label smoothing, 10% of the input.

(d) Label smoothing, 100% of the input.

Fig. 8 Dependence of the modulation recognition accuracy on the SNR level when using adversarial learning and label smoothing against various black-box attack algorithms with different adversarial perturbation budget limit values

has resulted in accuracy reduction in this case. However, label smoothing allows for improving the results in the case of the boundary and HopSkipJump attacks. In the beam selection use case, both adversarial training and label smoothing allow for some data rate increase in the case of the boundary and HopSkipJump attacks which are the most effective ones against the target AI/ML model in this use case.



(a) Adversarial learning, 10% of the input.

(b) Adversarial learning,100% of the input.

(c) Label smoothing, 10% of the input.

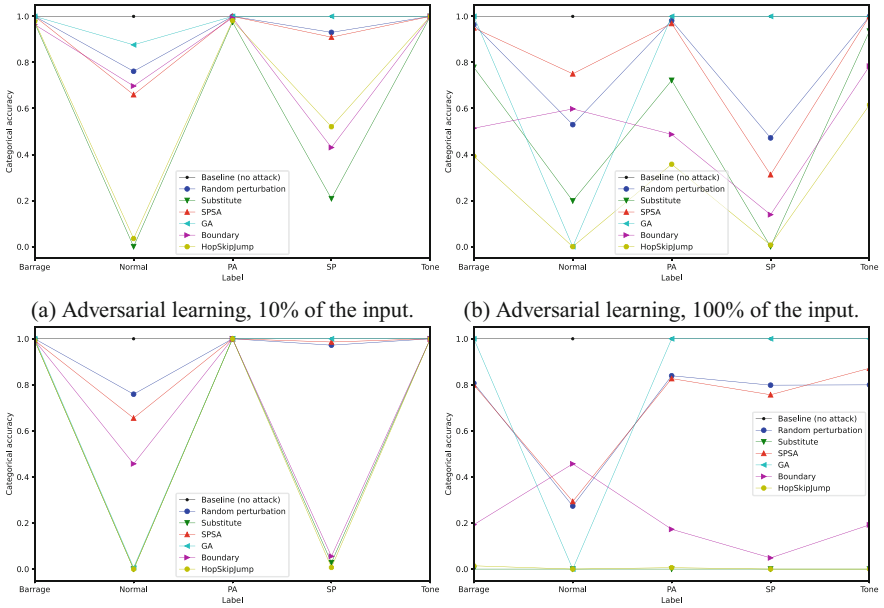
(d) Label smoothing, 100% of the input.

Fig. 9 Dependence of the sum-rate CDF on the beam selected based on the beam subset when using adversarial learning and label smoothing against various black-box attack algorithms with different adversarial perturbation budget limit values

Similarly to the previous use case, employing label smoothing worsens the network performance in the case of the substitute attack. Finally, in the jamming detection use case, adversarial training results in the detection accuracy increase in the worst-case scenario. However, as in two previous use cases, employing the substitute attack against the target model hardened with label smoothing allows the adversary to achieve 100% accuracy reduction in the case of higher perturbation budget values.

4.4 Result Discussion

As it has been shown in the first part of this section, adversarial examples generated with certain attack algorithms can have a devastating effect on the performance of intelligent mobile network components, especially when the perturbation budget is high enough. It is worth however mentioning that the results obtained in the beam selection use case are somewhat optimistic compared to the other two as in this case the adversary has to be able to produce the signal perturbation comparable in size with the original input of the target AI/ML model deployed; otherwise the negative effect of the attack is insignificant. This can be explained by the fact that



(a) Adversarial learning, 10% of the input.

(b) Adversarial learning, 100% of the input.

(c) Label smoothing, 10% of the input.

(d) Label smoothing, 100% of the input.

Fig. 10 Dependence of the jamming detection accuracy on the signal type when using adversarial learning and label smoothing against various black-box attack algorithms with different adversarial perturbation budget limit values

in this particular case the target model predicts the beam vector that results in the maximum achievable data rate, whereas the adversary aims to make the classifier to return some non-optimal beam. Despite being non-optimal, this vector may still result in some decent data rate. In theory, in order to reduce the network throughput significantly, the adversary may try to craft such a targeted perturbation that it not only causes a misclassification at the receiver's classifier, but also changes the beam to one of the worst vectors. However, more experiments are supposed to be carried out to test this hypothesis.

Speaking of the attack mitigation, the numerical experiments conducted have shown that the adversarial example detection and rejection approach can be employed when dealing with large signal perturbations in all three use cases studied. When the size of the adversarial perturbation is low, the situation is less optimistic for the network operator as most of the adversarial examples generated against the automatic modulation recognition and beam selection frameworks analysed would remain undetected. Finally, two model hardening techniques tested have not provided any noticeable improvements for the AI/ML models employed in all three use cases studied. On one hand, they allow for increasing classification accuracy slightly in the case of the most effective attack algorithms such as boundary and HopSkipJump. On the other hand, generating adversarial perturbations against the

target classifiers with the help of substitute models in the case of label smoothing has resulted in some noticeable accuracy decrease compared to the baseline case when no model hardening has been used. It is worth noticing that in our experiments we have assumed that the adversary is aware of the fact that a model hardening technique is employed and applies the same technique when training the substitute model which is not that unrealistic assumption in a real use case scenario.

5 Conclusion

In this study, we have evaluated various adversarial example generation attacks against machine learning models which can be deployed in future 5G networks for intelligent modulation recognition, beam selection, and jamming detection. First, we have summarised each of the problems formulated and discussed the data generation process. After that, the AI/ML model training and evaluation procedures have been overviewed. Next, multiple black-box attacks using various adversarial perturbation budget values have been employed against the target models and evaluated using the metrics selected. Finally, several adversarial example detection and model hardening techniques have been implemented and tested. Despite the significant negative impact the attacks discussed may achieve when employed against the target AI/ML-based 5G network components selected, unless there is a serious flaw in the component security, the adversary should be able to neither have access to the exact inputs of the target model, due to the different channel and interference conditions, nor obtain the output label, since it is most of the time used internally by the model and it is not available to any other wireless node outside of the network. For these reasons, in our future work, we are planning to focus on algorithms for crafting universal input-agnostic perturbations that can be generated by the adversary without having access to the model inputs and outputs. In particular, we are going to implement and test multiple such universal adversarial example generation algorithms as well as evaluate which detection and model hardening techniques can be employed to protect the models under attack.

References

1. Alkhateeb, A.: Deepmimo: A generic deep learning dataset for millimeter wave and massive MIMO applications. arXiv:1902.06435 (2019)
2. Alrabeiah, M., Alkhateeb, A.: Deep learning for TDD and FDD massive MIMO: Mapping channels in space and frequency. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, pp. 1465–1470. IEEE, Piscataway (2019)
3. Alrabeiah, M., Alkhateeb, A.: Deep learning for mmWave beam and blockage prediction using sub-6 GHz channels. IEEE Trans. Commun. **68**(9), 5504–5518 (2020)
4. Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C.J., Srivastava, M.B.: Genattack: Practical black-box attacks with gradient-free optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1111–1119 (2019)

5. Brendel, W., Rauber, J., Bethge, M.: Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. arXiv:1712.04248 (2017)
6. Cao, G., Lu, Z., Wen, X., Lei, T., Hu, Z.: Aif: an artificial intelligence framework for smart wireless network management. IEEE Commun. Lett. **22**(2), 400–403 (2018). https://doi.org/10.1109/LCOMM.2017.2776917
7. Catak, E., Catak, F.O., Moldsvor, A.: Adversarial machine learning security problems for 6G: mmWave beam prediction use-case. arXiv:2103.07268 (2021)
8. Chen, M., Saad, W., Yin, C., Debbah, M.: Echo state networks for proactive caching in cloud-based radio access networks with mobile users. IEEE Trans. Wirel. Commun. **16**(6), 3520–3535 (2017). https://doi.org/10.1109/TWC.2017.2683482
9. Chen, J., Jordan, M.I., Wainwright, M.J.: Hopskipjumpattack: A query-efficient decision-based attack. In: 2020 IEEE Symposium on Security and Privacy (SP), pp. 1277–1294. IEEE, Piscataway (2020)
10. Cousik, T.S., Shah, V.K., Erpek, T., Sagduyu, Y.E., Reed, J.H.: Deep learning for fast and reliable initial access in AI-driven 6G mmWave networks. arXiv:2101.01847 (2021)
11. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9185–9193 (2018). https://doi.org/10.1109/CVPR.2018.00957
12. Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B.: Detecting adversarial samples from artifacts. arXiv:1703.00410 (2017)
13. Fernández Maimó, L., Perales Gómez, A.L., Garcia Clemente, F.J., Gil Pérez, M., Martínez Pérez, G.: A self-adaptive deep learning-based system for anomaly detection in 5G networks. IEEE Access **6**, 7700–7712 (2018). https://doi.org/10.1109/ACCESS.2018.2803446
14. Forest, F., Lebbah, M., Azzag, H., Lacaille, J.: Deep embedded som: joint representation learning and self-organization. Reconstruction **500**, 500 (2000)
15. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv:1412.6572 (2015)
16. GSMA: Fs.31 - baseline security controls (2020)
17. GSMA: Fs.30 - security manual (2021)
18. Guo, C., Rana, M., Cisse, M., van der Maaten, L.: Countering adversarial images using input transformations. arXiv:1711.00117 (2018)
19. Guo, Q., Gu, R., Wang, Z., Zhao, T., Ji, Y., Kong, J., Gour, R., Jue, J.P.: Proactive dynamic network slicing with deep learning based short-term traffic prediction for 5G transport network. In: 2019 Optical Fiber Communications Conference and Exhibition (OFC), pp. 1–3 (2019)
20. Haidine, A., Salmam, F.Z., Aqqal, A., Dahbi, A.: Artificial intelligence and machine learning in 5G and beyond: a survey and perspectives. In: Moving Broadband Mobile Communications Forward: Intelligent Technologies for 5G and Beyond, p. 47 (2021)
21. Jin, H., Song, Q., Hu, X.: Auto-keras: An efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1946–1956 (2019)
22. Kim, B., Sagduyu, Y.E., Davaslioglu, K., Erpek, T., Ulukus, S.: Over-the-air adversarial attacks on deep learning based modulation classifier over wireless channels. arXiv:2002.02400 (2020)
23. Kim, B., Sagduyu, Y.E., Erpek, T., Davaslioglu, K., Ulukus, S.: Adversarial attacks with multiple antennas against deep learning-based modulation classifiers. arXiv:2007.16204 (2020)
24. Kim, B., Sagduyu, Y.E., Davaslioglu, K., Erpek, T., Ulukus, S.: Channel-aware adversarial attacks against deep learning-based wireless signal classifiers. arXiv:2005.05321 (2021)
25. Kim, B., Sagduyu, Y.E., Erpek, T., Ulukus, S.: Adversarial attacks on deep learning based mmwave beam prediction in 5G and beyond. arXiv:2103.13989 (2021)
26. Kim, B., Shi, Y., Sagduyu, Y.E., Erpek, T., Ulukus, S.: Adversarial attacks against deep learning based power control in wireless communications. arXiv:2109.08139 (2021)
27. Kohonen, T.: Self-Organizing Maps, vol. 30. Springer Science & Business Media, Cham (2012)

28. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv:1607.02533 (2017)
29. Liu, Q., Guo, J., Wen, C.K., Jin, S.: Adversarial attack on DL-based massive MIMO CSI feedback. J. Commun. Netw. **22**(3), 230–235 (2020). https://doi.org/10.1109/JCN.2020.000016
30. Lyu, W., Zhang, Z., Jiao, C., Qin, K., Zhang, H.: Performance evaluation of channel decoding with deep neural networks. In: 2018 IEEE International Conference on Communications (ICC), pp. 1–6 (2018). https://doi.org/10.1109/ICC.2018.8422289
31. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv:1706.06083 (2019)
32. Manoj, B.R., Sadeghi, M., Larsson, E.G.: Adversarial attacks on deep learning based power allocation in a massive MIMO network. arXiv:2101.12090 (2021)
33. Masri, A., Veijalainen, T., Martikainen, H., Mwanje, S., Ali-Tolppa, J., Kajó, M.: Machine-learning-based predictive handover. In: 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 648–652 (2021)
34. Meng, D., Chen, H.: Magnet: a two-pronged defense against adversarial examples. arXiv:1705.09064 (2017)
35. Meng, F., Chen, P., Wu, L., Wang, X.: Automatic modulation classification: a deep learning enabled approach. IEEE Trans. Vehic. Technol. **67**(11), 10760–10772 (2018). https://doi.org/10.1109/TVT.2018.2868698
36. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. arXiv:1702.04267 (2017)
37. Minovski, D., Ogren, N., Ahlund, C., Mitra, K.: Throughput prediction using machine learning in LTE and 5G networks. IEEE Trans. Mob. Comput., 1–1 (2021). https://doi.org/10.1109/TMC.2021.3099397
38. Nicolae, M.I., Sinn, M., Tran, M.N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I.M., Edwards, B.: Adversarial robustness toolbox v1.0.0. arXiv:1807.01069 (2019)
39. O'Shea, T., West, N.: Radio machine learning dataset generation with GNU radio. Proc. GNU Radio Conf. **1**(1) (2016). https://pubs.gnuradio.org/index.php/grcon/article/view/11
40. O'Shea, T.J., Corgan, J., Clancy, T.C.: Convolutional radio modulation recognition networks. In: International Conference on Engineering Applications of Neural Networks, pp. 213–226. Springer, Berlin (2016)
41. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. arXiv:1511.07528 (2015)
42. Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Kurakin, A., Xie, C., Sharma, Y., Brown, T., Roy, A., et al.: Technical report on the cleverhans v2. 1.0 adversarial examples library. arXiv:1610.00768 (2016)
43. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A.: Practical black-box attacks against machine learning. arXiv:1602.02697 (2017)
44. Pawlak, J., Li, Y., Price, J., Wright, M., Al Shamaileh, K., Niyaz, Q., Devabhaktuni, V.: A machine learning approach for detecting and classifying jamming attacks against OFDM-based uavs. In: Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning, pp. 1–6 (2021)
45. Peng, S., Jiang, H., Wang, H., Alwageed, H., Yao, Y.D.: Modulation classification using convolutional neural network based deep learning model. In: 2017 26th Wireless and Optical Communication Conference (WOCC), pp. 1–5 (2017). https://doi.org/10.1109/WOCC.2017.7929000
46. Peng, B., Seco-Granados, G., Steinmetz, E., Fröhle, M., Wymeersch, H.: Decentralized scheduling for cooperative localization with deep reinforcement learning. IEEE Trans. Vehic. Technol. **68**(5), 4295–4305 (2019). https://doi.org/10.1109/TVT.2019.2913695
47. Rauber, J., Brendel, W., Bethge, M.: Foolbox: A python toolbox to benchmark the robustness of machine learning models. arXiv:1707.04131 (2018)

48. Ross, A.S., Doshi-Velez, F.: Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. arXiv:1711.09404 (2017)
49. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International Conference on Machine Learning, pp. 4393–4402. PMLR (2018)
50. Sadeghi, M., Larsson, E.G.: Physical adversarial attacks against end-to-end autoencoder communication systems. arXiv:1902.08391 (2019)
51. Safari, M.S., Pourahmadi, V., Sodagari, S.: Deep UL2DL: data-driven channel knowledge transfer from uplink to downlink. IEEE Open J.f Vehic. Technol. **1**, 29–44 (2020). https://doi.org/10.1109/OJVT.2019.2962631
52. Sagduyu, Y.E., Erpek, T., Shi, Y.: Adversarial machine learning for 5G communications security. arXiv:2101.02656 (2021)
53. Sanguinetti, L., Zappone, A., Debbah, M.: Deep learning power allocation in massive MIMO. arXiv:1812.03640 (2019)
54. Schwarzmann, S., Marquezan, C.C., Trivisonno, R., Nakajima, S., Zinner, T.: Accuracy vs. cost trade-off for machine learning based QoE estimation in 5G networks. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020). https://doi.org/10.1109/ICC40277.2020.9148685
55. Shi, Y., Sagduyu, Y.E.: Adversarial machine learning for flooding attacks on 5G radio access network slicing. arXiv:2101.08724 (2021)
56. Shi, Y., Sagduyu, Y.E., Erpek, T.: Reinforcement learning for dynamic resource optimization in 5G radio access network slicing. In: 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–6 (2020). https://doi.org/10.1109/CAMAD50429.2020.9209299
57. Shi, Y., Sagduyu, Y.E., Erpek, T., Gursoy, M.C.: How to attack and defend 5G radio access network slicing with reinforcement learning. arXiv:2101.05768 (2021)
58. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Advances in Neural Information Processing Systems, vol. 25 (2012)
59. Sotgiu, A., Demontis, A., Melis, M., Biggio, B., Fumera, G., Feng, X., Roli, F.: Deep neural rejection against adversarial examples. arXiv:1910.00470 (2020)
60. Steinhardt, J., Koh, P.W., Liang, P.: Certified defenses for data poisoning attacks. arXiv:1706.03691 (2017)
61. Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X., Sidiropoulos, N.D.: Learning to optimize: training deep neural networks for interference management. IEEE Trans. Signal Process. **66**(20), 5438–5453 (2018). https://doi.org/10.1109/TSP.2018.2866382
62. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv:1312.6199 (2014)
63. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
64. Uesato, J., O'Donoghue, B., van den Oord, A., Kohli, P.: Adversarial risk and the dangers of evaluating against weak attacks. arXiv:1802.05666 (2018)
65. Usama, M., Mitra, R.N., Ilahi, I., Qadir, J., Marina, M.K.: Examining machine learning for 5G and beyond through an adversarial lens. arXiv:2009.02473 (2020)
66. Wang, F., Gursoy, M.C., Velipasalar, S.: Adversarial reinforcement learning in dynamic channel access and power control. arXiv:2105.05817 (2021)
67. Wang, H., Miller, D.J., Kesidis, G.: Anomaly detection of adversarial examples using class-conditional generative adversarial networks. Comput. Secur. **124**, 102956 (2023)
68. Wen, C.K., Shih, W.T., Jin, S.: Deep learning for massive MIMO CSI feedback. IEEE Wirel. Commun. Lett. **7**(5), 748–751 (2018). https://doi.org/10.1109/LWC.2018.2818160
69. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. In: Proceedings 2018 Network and Distributed System Security Symposium (2018). https://doi.org/10.14722/ndss.2018.23198.

70. Zantedeschi, V., Nicolae, M.I., Rawat, A.: Efficient Defenses Against Adversarial Attacks, pp. 39–49. Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3128572.3140449
71. Zolotukhin, M., Miraghaei, P., Zhang, D., Hämäläinen, T.: On assessing vulnerabilities of the 5G networks to adversarial examples. IEEE Access **10**, 126285–126303 (2022)
72. Zolotukhin, M., Miraghaie, P., Zhang, D., Hämäläinen, T., Ke, W., Dunderfelt, M.: Black-box adversarial examples against intelligent beamforming in 5G networks. In: 2022 IEEE Conference on Standards for Communications and Networking (CSCN), pp. 64–70. IEEE, Piscataway (2022)

Designing and Implementing an Interactive Cloud Platform for Teaching Machine Learning with Medical Data

Frederic Jonske, Kevin Osthues, Amin Dada, Enrico Nasca, Jana Fragemann, Julian Alff, Oleh Bakumenko, Marcel Birnbach, Maxim Kondratenko, Lars Reinike, Benjamin Schulz, Fabian Siethoff, Tobias Simon, Joey Wang, Nils Zhang, Fin H. Bahnsen, Jan Egger, Moon-Sung Kim, Maria Lymbery, Jens Kleesiek, and Johannes Kraus

1 Introduction

Machine learning (ML) is a data-driven approach for teaching computers to perform certain tasks, where the decision process itself is "learned" from data using some algorithm, with only a few (if any) explicit rules provided by humans. The concepts of machine learning and, more generally, artificial intelligence (AI) have been around since the 1950s, almost as long as computers themselves [66]. While the field experienced multiple periods of decreased research interest due to its perceived conceptual limitations and hardware constraints [31], the advent of modern machine learning concepts and the steadily increasing computational power and availability of GPUs, affordable even for amateur enthusiasts, has seen the field go through explosive growth in terms of research, application, and public

F. Jonske (✉) · A. Dada · E. Nasca · J. Fragemann · F. H. Bahnsen · J. Egger · M.-S. Kim · J. Kleesiek
Institute of AI in Medicine (IKIM), University Medicine Essen, Essen, Germany
e-mail: frederic.jonske@uk-essen.de
https://www.ikim.uk-essen.de/; amin.dada@uk-essen.de; enrico.nasca@uk-essen.de; jana.fragemann@uk-essen.de; fin.bahnsen@uk-essen.de; jan.egger@uk-essen.de; moon-sung.kim@uk-essen.de; jens.kleesiek@uk-essen.de

K. Osthues · J. Alff · O. Bakumenko · M. Birnbach · M. Kondratenko · L. Reinike · B. Schulz · F. Siethoff · T. Simon · J. Wang · N. Zhang · M. Lymbery · J. Kraus
Faculty of Mathematics, University Duisburg-Essen, Essen, Germany
e-mail: kevin.osthues@uni-due.de
https://www.uni-due.de/mathematik/; julian.alff@stud.uni-due.de; oleh.bakumenko@stud.uni-due.de; marcel.birnbach@stud.uni-due.de; maxim.kondratenko@stud.uni-due.de; lars.reinike@stud.uni-due.de; benjamin.schulz@stud.uni-due.de; fabian.siethoff@stud.uni-due.de; tobias.simon@stud.uni-due.de; joey.wang@stud.uni-due.de; nils.zhang@stud.uni-due.de; maria.lymbery@uni-due.de; johannes.kraus@uni-due.de

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_12

interest [72], particularly during the past decade. Deep learning (DL) is a subset of ML that sacrifices some of the explainability of the often partially handcrafted ML models in favor of larger, generally intractable, decision processes. The latter offer greatly increased performance on downstream tasks and require minimal human input beyond the initial setup to train themselves. DL models, in particular so-called artificial neural networks, named for their functional similarity to the human brain, have been both the primary driver and recipient of this increased interest.

For decades already ML has found important applications in the field of medicine [31], e.g., in enhancing diagnosis and detection [63], segmentation [16, 25], survival prognosis [32], surgical planning [34], personalized treatment [64], and the discovery of biomarkers for specific pathologies [26], as clinical data management tools [30], or as research tools [15, 38, 52, 62]. The predictive and analytical power of these ML models has grown steadily and now decisions based on it are qualitatively competitive with (and often more efficient than) those made by healthcare professionals [14], occasionally even outperforming them on specific, narrowly defined tasks such as pneumonia detection in chest X-rays [53], or improving radiologists' performance during breast cancer screenings [71].

While the development of its subfields is responsible for this progress, DL has been met with considerable skepticism and apprehension and its practical application in real clinical settings has been almost nonexistent [14]. This is not only because of its lack of explainability but also due to ethical concerns such as data privacy or implicit racial biases. In a more general context and for similar reasons, a significant fraction of ML research in medicine remains focused on canonical ML (e.g., random forests [9], boosted decision trees [41], support vector machines [11]) methods over more modern DL approaches (e.g., convolutional neural networks [19, 24, 47], U-Nets [28, 56], or vision transformers [13]).

The state of the art of AI in medicine is continually driven forward. If the goal for AI tools is to eventually incorporate them into the clinical routine, the demand for professionals with degrees in ML or medicine and a strong foundation in the respective other discipline is likely to increase. In order for this integration process to be sustainable and safe, higher literacy in the concepts of ML and DL across traditionally separate professions is required. The more comprehensive the understanding of the various strengths and weaknesses of DL is, the more safely can this technology be applied.

In this spirit, we present our experience with designing a seminar entitled "Machine Learning in Medicine - Theory & Practice." The course was on an introductory level and has been offered to master students in mathematics with the goal of building and understanding state-of-the-art neural network architectures using PyTorch [49], as well as their strengths, weaknesses, and points of failure. The respective models have been tested and evaluated on the medical imaging challenge dataset LiTS 2017 (Liver Tumor Segmentation) [8], a dataset containing abdominal computed tomography (CT) scans. The choice of the architectures covered by the seminar was inspired by the milestone advances during the last decade of DL research, with a focus on the medical imaging perspective, and included sections on AlexNet [35], ResNet [24], U-Net [56], and the Vision Transformer [13]. The

seminar was assigned nine credit points according to the European Credit Transfer and Accumulation System (ECTS) and jointly offered by the chairs of Numerical Mathematics, Faculty of Mathematics, and the Institute for AI in Medicine (IKIM), University Hospital of Essen at the University of Duisburg-Essen, over the span of one semester. The seminar could be attended both in person and remotely. We aimed to address the following research/engineering questions: (a) Can a hands-on machine learning seminar that contains real-world challenges be approachable to students with only limited previous experience in machine learning and programming expertise? (b) Can it be implemented entirely as an uncomplicated, out-of-the-box learning platform using only cloud-based computation resources?

Although the seminar was offered to master students at the Faculty of Mathematics with varying prior knowledge of the mathematical foundations of ML and DL, the lessons themselves are designed to be appropriate for amateurs with a low or medium level of programming skill, regardless of study direction. In addition to discussing the learning objectives and practical challenges of the seminar, we make all of the discussed content available publicly, including student-created sample solutions for each task.

Furthermore, we explore the technical design and specifications of the interactive cloud platform which is based entirely on external resources, such as Coder [6] and AWS [48], and requires no dedicated hardware beyond hosting a Web page. The platform is designed to streamline access to the learning material by both students and tutors, with tutors having live access to student workspaces to facilitate uncomplicated remote debugging. This live access for tutors was equivalent to the access which students had; tutors could see all code in their students' workspaces, observe the execution or start it themselves, and even comment or edit code concurrently with the students. We share our codebase and instructions for setting up the cloud platform [44] along with materials [29] on GitHub. While all course materials benefit from execution on dedicated hardware, the entirety of the seminar can also be completed using only a modern PC.

Finally, we discuss the results of the seminar, reflect on the lessons we have drawn from our experience in the context of our abovementioned research questions, and conclude our contribution with an outlook on AI in medicine and education and its role in an increasingly digitized world.

2 Design Principles

Before we dive into the technical realization or the actual learning materials, we want to motivate some of the design principles of both the platform and the learning materials of our seminar. Some of the ideas we present below are very specific to the field of ML, while others are generally applicable to any subject taught remotely or in a hybrid fashion.

2.1 Teaching Requirements

Every project, particularly programming projects, comes with primary and secondary challenges. Primary challenges concern implementing an idea effectively and solving subject-related questions along the way. The secondary challenges relate to the preparation or maintenance of the work environment. Typically, one is only interested in the former, and the latter are accepted as necessary. When the secondary challenges significantly outweigh the primary ones, or even outright prevent working on them, we encounter what we term "logistical frustration." If this frustration is strong enough, it can cause a project to be pursued with reduced interest or even abandoned.

Logistical frustration is a common problem in computer science because of the impenetrable pile of layered dependencies, but is absolutely not limited to it. If driving your car required repairing it every other day, you would quickly reconsider having a car. Hence, our first and foremost design principle is to reduce the potential for logistical frustration among the students to a minimum. In the best-case scenario, any problem the students have is only related to their understanding and implementation of the subject matter. From this premise follow several requirements:

- The students should be able to work on the course materials from any machine and any operating system, indicating the need for Web-based access.
- Connecting to and navigating the space in which the course materials are made available should be quick and should not require a high degree of technical literacy.
- Setting up the environment in which the students work should be quick, easy, and uniform across students. Hence, we decided to prepare containerized environments (see Sect. 3) ahead of time so that the students would not need to interact with them.
- The same principle applies to some degree to the code students work on. Wherever possible, code not immediately relevant to the task at hand should be provided by us as a sort of guard rail, both for simplicity and uniformity.
- Any such guard rail code should be static, as any evolution of this code parallel to efforts by the students that rely on it can introduce problems the students cannot (or not easily) fix themselves.

Note that while creating and managing your programming environment is a highly important skill, we opted not to try and confer it to students during our seminar for multiple reasons: Firstly, as the entirety of the course would only require a single environment, students were unlikely to internalize the skill due to it seeing little use. Secondly, the centrally managed environment offers scalability with respect to the number of students. This scalability is lost when every student creates their own environment, which invariably ends up being unique in some unforeseen aspect and introduces additional challenges during debugging down the line.

2.2 Interactivity

It is generally accepted that learning by doing something is far more effective than learning by simply reading or hearing about something [7, 18] and this is particularly true when it comes to learning programming (or a specific programming skill) [40]. Hence, we aim to include the relevant programming tasks with a preamble as concise as possible and to encourage the students to learn to navigate through the run-debug cycle on their own. To this end, any code which the students interact with is provided as the Jupyter Notebook [33], a readable and highly interactive programming environment for Python code that allows for partial code execution and easy debugging.

2.3 Pair Programming and Teams

Pair programming is an established practice [10] in agile software development [12], wherein two programmers work on a piece of code together. Classically, this is done in front of a single machine, but modern developments allow the concept to be extended to two users remotely accessing the same piece of code in a shared coding session.

In addition to the typical benefits of increased code quality and development speed [4, 10], assigning the students to pairs or teams has some pedagogically valuable consequences as well. For one, the students learn to express their ideas in terms of technical jargon when communicating with each other. Additionally, if the teams are arranged appropriately, the students with greater initial knowledge can share their knowledge with those who have less prior experience. Student satisfaction has been shown to be increased among students working in pairs compared to those working solo [61].

Finally, student teams are valuable from an organizational perspective. Offering help to the teams during lessons drastically reduces the number of problems that need to be fixed simultaneously, while simplifying the process of reviewing the student's solutions. Consequently, the formation of teams was chosen as a cornerstone of our seminar.

2.4 Teaching Methods

In terms of teaching methods, we attempt to integrate lessons learned from [65]. As tutors with domain knowledge in informatics, mathematics, and medicine, we can best deliver domain specificity in the field of medical DL, which partly substantiates our later choice of dataset (see Sect. 4.3.4). In addition to this, we attempt to integrate as many real-world examples as possible into our explanations,

for example, exploring the concept of overfitting to training data for different reasons or implicit biases (and the corresponding solutions) on real failure cases. Note that there is a number of comments and instructions in the learning materials addressing this problem; however, detailed explanations were given in specific discussions with the students during the seminar.

We worked to include the ideas of constructive alignment [7] into our teaching strategies. The alignment of learning objectives, pedagogical practices, and evaluation systems is emphasized by constructive alignment. We constructed our teaching strategies to make it easier to attain these objectives by stating the targeted learning outcomes explicitly. Implementing this strategy, we focus on actively involving students in the learning process and giving them the chance to develop their knowledge and abilities. Furthermore, we can precisely gauge students' development and comprehension by matching our evaluation techniques with the learning objectives. This aids in establishing a harmonious and efficient learning atmosphere that encourages our students' engaging learning experiences.

3 Technical Implementation

In order to attend lessons and complete coursework, students require access to compute hosts equipped with data center-grade graphics processing units (GPUs). These resources are traditionally found in high-performance computing clusters where users are expected to authenticate using a secure shell (SSH) client and interact with GNU/Linux hosts via a command-line interface. In recent years, the technology industry has seen a growth of remote development environment solutions hosted on cloud providers [21, 43, 54] or even on-premises [22, 70]. These services allow software developers to access pre-configured environments using a Web browser with the option to attach a local integrated development environment (IDE). We ourselves have some preliminary work in this direction which we describe in [4, 5]. While typically aimed at improving the productivity of software engineers by eliminating sources of friction, they have the potential for lowering the complexity barrier that discourages learners or people from different areas of expertise from using specialized computing resources.

In this section, we discuss the requirements and implementation of a remote development platform that supports a hands-on machine learning course without requiring a strong level of computer literacy. We aimed for a vendor-neutral design as we prefer exposing students to general workflows rather than proprietary solutions whenever possible, and allowed the implementation to include proprietary components only as long as they could be replaced without a cascade effect on the other elements. The choice for each individual service was informed by the desired features or guarantees we explored in Sect. 2 and the technical requirements of the following subsection.

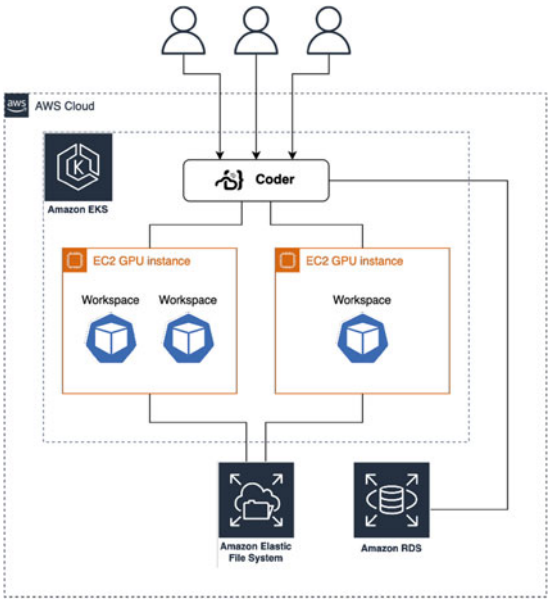


Fig. 1 Deployment of the work environments. Students are only ever exposed to the Web interface from which they can start and connect to their workspace(s) with just two button clicks

Our implementation is based on Coder (cf. Sect. 3.2) for the main service and Amazon Web Services (AWS) for hosting and ancillary services (cf. Sect. 3.4). The architecture is depicted in Fig. 1. To avoid clutter, components of secondary relevance such as DNS are omitted. Our entire deployment material is available at [44].

3.1 Technical Requirements

The computational resource provisioning was designed according to the following principles:

- There should be enough computation resources for all students to log on and work in parallel.
- Given that GPUs are generally a scarce resource in relation to the number of users, it is necessary to devise a mapping strategy. Ideally, there would be an automated mechanism to assign two or more "seats" per GPU. As a fallback, we considered simply dividing the GPUs equally among students during the registration phase in a manual fashion.
- Access to the platform should be available from anywhere and at any time, and students should be able to resume previous sessions, to maximize the students' productivity between lessons. As an additional benefit, this design also allows our seminar to be held in a hybrid fashion.

- The same availability criterion can be extended to the teaching staff. The more easily we can access the students' code, the more quickly and effectively we can help them with debugging. By allowing direct access to students' workspaces, we reduce wasted time during and expedite problem-solving outside of lessons.

In addition, the platform should be optimized for robustness from the viewpoint of the students:

- The system should be robust to crashes. If the machine that a student is working on crashes, the student should only have to restart, not wait until the responsible admins have finished maintenance.
- The system should be robust to loss of data. If the storage and computational resources are separated, and storage is backed up in some reliable fashion, then the failure of any component has minimal consequences for the student, beyond having to redo the last few minutes of coding. Hence, we designed the storage available to the students as persistent while simultaneously encouraging backing up data externally.

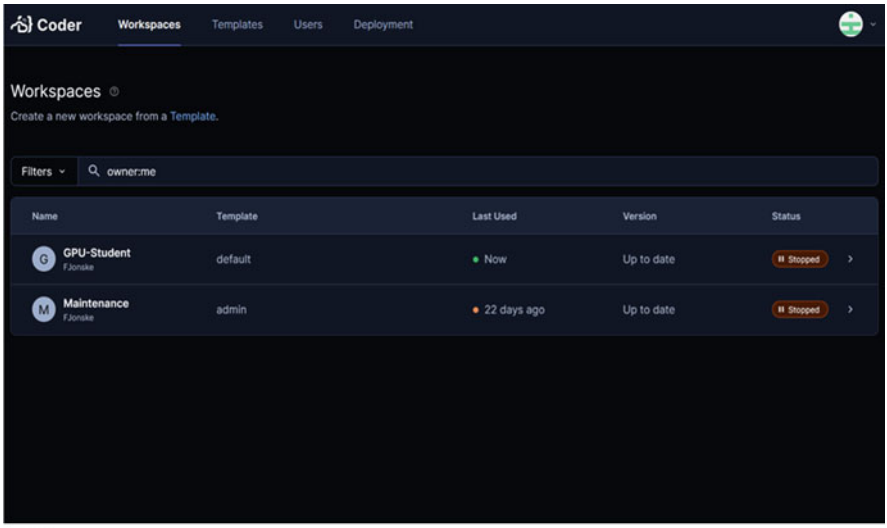
3.2 Coder

Coder is a development environment engine by Coder Technologies. The company offers an open-source release under the GNU AGPLv3 license and an Enterprise upgrade with additional features. Our deployment adopts the open-source release.

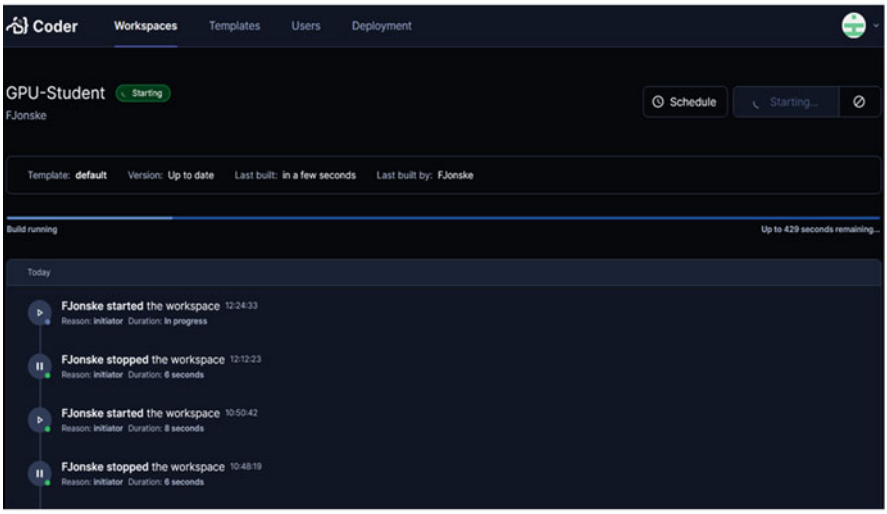
A minimal installation consists of the Coder server and a PostgreSQL database deployed separately. Users can interact with the server via a Web interface or by using the Coder client application. By connecting to the Coder server, users can create and manage development environments called workspaces. As a distinctive feature, Coder does not perform the task of creating the workspace directly; instead, it hands the responsibility over to Terraform. Administrators provide their own Terraform configuration (called templates in Coder), thus allowing complete control over the creation of workspaces.

As previously touched on (cf. Sect. 2.1), we created a template that implements containerized workspaces. Containers are created from a custom Docker [42] image pre-populated with all the necessary software for tutorials and hands-on lessons. Additionally, the template installs a Web-based variant of Visual Studio Code [69]. This setup enables a workflow that can be carried out entirely via a Web interface (Fig. 2): (a) sign into Coder, (b) click a button to create a workspace, (c) click a button to launch a Visual Studio Code window, (d) work on the actual assignments, and (e) at the end of the session, click a button to stop the workspace and sign out. Data stored in the home directories of a workspace is preserved between sessions for each individual workspace. For administrators, every student workspace is accessible in the same way, even while in use, facilitating easy remote support or

submission of solutions. This access is fully privileged, with administrators able to read, modify, and execute everything in the workspace, even concurrently, with tutor and students viewing and editing the same file.

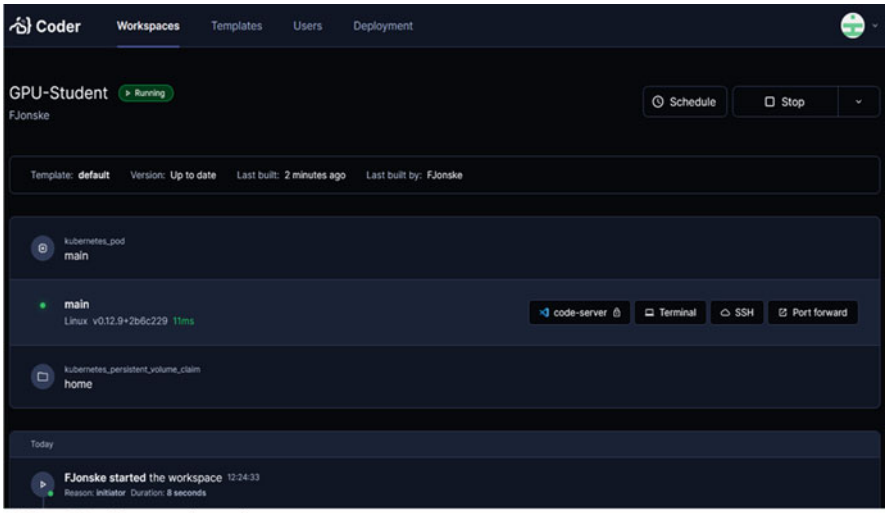


(a) Workspace selection and creation

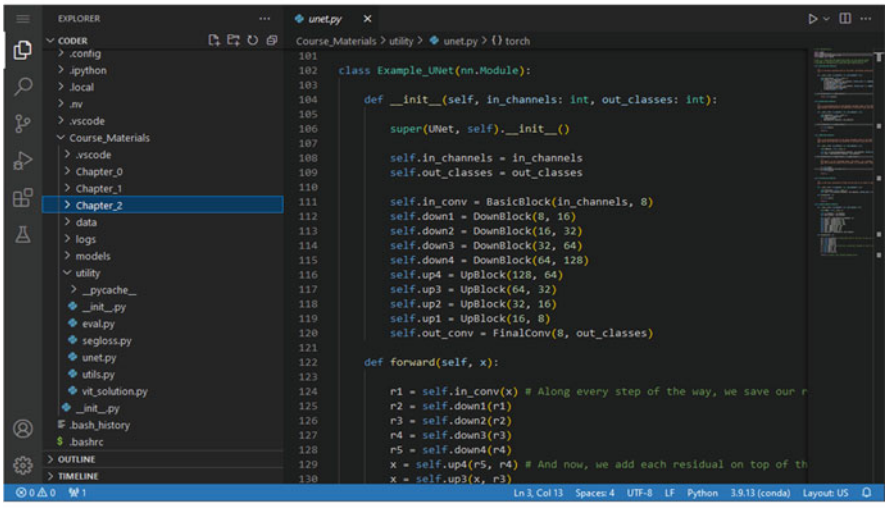


(b) Automatic container start-up and resource provisioning

Fig. 2 UI of the interactive cloud platform. After logging in, only a single click is required to get from panel to panel. The creation and initialization of the containerized environment, resource provisioning, and the Code Server itself are handled in an entirely automated fashion and require no input from the students. Connection options to the workspace are user-friendly and simple. The coding environment closely resembles Visual Studio Code [69]



(c) User-friendly connection options



(d) VSCode-style coding environment

Fig. 2 (continued)

3.3 Kubernetes

Workspaces were implemented as Kubernetes pods. By deploying Cluster Autoscaler [36] and NVIDIA GPU Operator [46] on Kubernetes, we were able to tackle key issues with relatively little effort. One such issue is the operating cost associated with data center GPUs, especially when hosted by cloud providers due to the high demand for specialized hardware. We set out to make optimal use of the available resources.

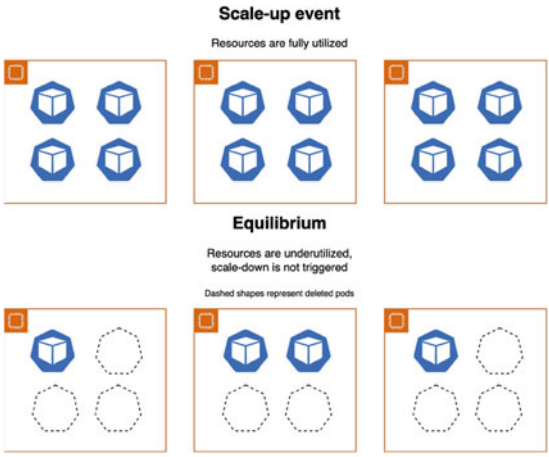


Fig. 3 Hypothetical scale-up scenario in which 3 nodes were required to satisfy the peak resource demand of 12 students. Note that multiple nodes remain active unnecessarily after most students have left because evictions only happen once nodes are entirely empty. This scenario did not occur in practice, as the demand typically only spiked during the seminar itself, after which every student would log off at least until they were home

When a Kubernetes pod specification includes a request for one or more GPUs, the native scheduling mechanism assigns the desired amount of GPUs for that pod exclusively. This approach may be inefficient if the individual GPUs are not fully utilized. Since we can calculate the amount of GPU RAM that each student is supposed to allocate to complete the exercises, we can determine whether a GPU could accommodate more than one pod. Let R be the required GPU RAM for a pod and G the installed GPU RAM; if

$$\lfloor G/R \rfloor = n, \quad n \geq 2$$

holds, then the GPU can support n concurrent users. The NVIDIA GPU Operator provides a time-slicing feature that exposes multiple logical devices for each physical GPU, thereby allowing pods to share access to the same underlying hardware.

Cluster Autoscaler is used in concert with GPU time-slicing to ensure that only the required amount of cloud nodes are active at any given time. When the first workspace is launched, a node is scaled up. As more workspaces are launched, they will exhaust the GPU capacity of the node and the autoscaler will scale up another node. The autoscaler is also able to scale down a node after all workspaces scheduled on it have been stopped. With this mechanism in place, only nodes with active workspaces contribute to costs.

An undesirable situation depicted in Fig. 3 may arise in the following scenario: (a) a demand spike triggers a significant scale-up event, (b) a subset of workspaces evenly distributed across nodes are stopped, and (c) the system enters an equilibrium such that workspaces are equally started and stopped over time, never meeting the scale-down condition. In a deployment with stateless pods, the problem is resolved by allowing the autoscaler to evict pods and reschedule them on different nodes to achieve optimal usage. However, in a context where pods serve as interactive workspaces, it is not reasonable to allow unprompted eviction.

We did not observe this problem in practice, although it might become noticeable with a large number of users. As a mitigation strategy for concepts based on our work, many nodes equipped with few GPUs each should be preferred to few nodes with many GPUs each. Additionally, workspaces should have a shutdown timer to avoid keeping them active indefinitely.

3.4 Deployment

The Coder server and workspaces were deployed on Amazon Elastic Kubernetes Service [2]. A PostgreSQL database for Coder was created using Amazon Relational Database Service [3]. To allow for preserving important data when a workspace is stopped, the Coder template configures the home directory as an Network File System (NFS) mount point. Amazon Elastic File System [1] was used as a storage server exposing NFS shares for home directories. If a workspace is deleted, the associated NFS share is deleted as well. It should be noted that our design does not depend strictly on AWS and can be replicated using any cloud provider or by hosting it using on-premise resources.

A drawback of relying on AWS-hosted GPUs for hands-on classes is the scarcity of specialized hardware: a request for a GPU node – either manually or via autoscaling – may not be met due to unavailability at that particular time. An additional hindrance involves the approval process for quotas for specialized hardware. Each quota is tied to a specific instance type and cannot be transferred to a different instance type. As a result, if a scale-up request is denied, launching a different instance type can be attempted only if another quota has been approved ahead of time. The quota approval process hampers on-the-fly experimentation and adaptation. It is therefore crucial to prepare well in advance and estimate the hardware requirements as closely as possible, particularly in relation to costs.

3.5 Alternative Implementation Approaches

Multiple alternatives to the learning platform we designed exist, each of them with their own advantages and disadvantages. In the following, we will give a short overview over these approaches and our reason behind foregoing them. A post hoc assessment of our learning platform that largely mirrors our initial expectations can be found in the Discussion section.

Coder Platform with On-premise Resources Our learning platform can theoretically be hosted on any collection of compute hardware, given some small adjustments. The advantages of hosting the learning platform on on-premise resources are absolute control over all available resources, reduced costs compared to rented cloud compute resources, and the reuse of existing infrastructure to manage users

and access. Disadvantages include the necessity of pre-existing compute resources and their regular availability, both regarding other users and potential downtimes of the local system and the added workload of maintaining said system. In practice, we considered the guaranteed uptime of the platform and guaranteed availability of compute resources a high priority, and the implementation of a platform that would work with cloud resources helpful for the wider the community.

On-premise Hosting Without Coder Platform One alternative to the learning platform approach would be to have students directly access on-premise compute resources and work there. The obvious advantage here would be that no platform must be developed or maintained. However, we considered the disadvantages of the students having to familiarize themselves with the environment of the local high-performance compute cluster and its resource allocation and environment management to outweigh the gains. Additionally, we expected the time cost debugging of any student solutions (or even problems that have nothing to do with the learning materials themselves) to increase compared to our approach – the more variables are left in the hands of the students, the more slightly different problems invariably end up happening.

Distributed Learning Materials Finally, all learning materials can be distributed to students and executed individually on their own machines. This approach was strongly disfavored by us due to the compute resource requirements of deep learning generally outstripping the power of modern personal computers, and laptops in particular. Further, this approach fosters unique environments, code editors, and operating systems combinations, which in turn exacerbate the strain on tutors, as every instance of debugging takes longer in this scenario.

4 Course Organization and Materials

In the following section, we detail the subject of the individual lessons, the style of these lessons, and their specific learning objectives and discuss the provided guard rail code and grading guidelines we employed.

4.1 Organization

As mentioned in Sect. 1, we loosely followed the history of machine learning, specifically the topic of image classification and segmentation using neural networks and their applications in medicine, when planning the lessons. Consequently, we split the semester into 6 topically distinct blocks, each containing several lessons of 180 minutes spaced in 1-week intervals. The first two of those blocks were reserved for introductory lessons, while the remaining four were engaged with one neural network architecture each that we deemed evolutionary steps, both in general ML

and specifically in the context of ML in medicine. Our choice fell on AlexNet, ResNet, U-Net, and the Vision Transformer (ViT).

In accordance with the principles outlined in Sect. 2.3, we divided the ten students into four groups of two or three students, respectively, allowing the students the choice of group constellations and presented topics, but encouraging the formation of teams in which members had different skill levels (ranging from no prior experience in Python to significant prior experience in both Python and PyTorch; see Fig. 5). Building on the thoughts of Sect. 2.2, we chose to let the lessons themselves be partially conducted by the student teams, each of which familiarized itself with one of the core topics (see Sect. 4.5) of the course, presenting the subject matter before starting the respective practical section. This approach is also known as flipped classroom [45] or inverted classroom and has been reported to increase the motivation and engagement with learning materials, as well as the effectiveness of the learning process.

Each block then featured a programming task including the implementation of the discussed architecture, which every student had to complete. The block concluded with a presentation by the group that presented the architecture, in which they explained their solution and showcased their results. This final presentation was held at the beginning of the first lesson of a new block, marking the latest point at which solutions for that task could be handed in. As the concepts behind the U-Net and segmentation, in general, and the Vision Transformer were more complicated than those behind AlexNet and ResNet, we assigned the larger groups to the former topics. An account of the allocated times for the individual blocks is illustrated in Table 1.

All lessons were held in a hybrid format, giving students the choice of showing up in person and with any portable machine capable of displaying a Web page or joining the session remotely via video call, if they were unable to attend in person. Portable machines with common software setups were available for students to use in case of unforeseen software incompatibilities, but ended up not being used as all students were able to access and work on the course materials from their various own devices.

Table 1 Seminar timetable. Organizational and theoretical sections are colored in green, presentations in blue, and implementation sections in orange

Week 1	Organisational	Introduction to Python	Break	Introduction to Python	
Week 2	Introduction to Python		Break	Introduction to Python	GitHub Tutorial
Week 3	PR by tutors: Neural Networks	Introduction to PyTorch	Break	Introduction to PyTorch	Teams assigned
Week 4	Introduction to PyTorch		Break	Introduction to PyTorch	
Week 5	PR Team 1: AlexNet	AlexNet implementation	Break	AlexNet implementation	
Week 6	AlexNet implementation		Break	AlexNet implementation	
Week 7	PR Team 1: AlexNet solution	PR Team 2: ResNet	Break	ResNet implementation	
Week 8	ResNet implementation		Break	ResNet implementation	
Week 9	PR Team 2: ResNet solution	PR Team 3: U-Net	Break	U-Net implementation	
Week 10	U-Net implementation		Break	U-Net implementation	
Week 11	U-Net implementation		Break	U-Net implementation	
Week 12	PR Team 3: U-Net solution	PR Team 4: ViT	Break	ViT implementation	
Week 13	ViT implementation		Break	ViT implementation	
Week 14	PR Team 4: ViT solution	ViT implementation	Break	ViT implementation	

4.2 Grading Guideline

While we encouraged completing the tasks as a group, the submission of the solution itself was strictly individual. During the first lesson blocks, each student would present their solution privately to two tutors, so that the tutors would learn to apply a common grading standard (cf. Sect. 4.2). The submissions during later lessons were reviewed only by one tutor. Any such submission was a dialogue between the tutor(s) and the student, similar to an oral exam. During these dialogues, we occasionally asked questions, trying to guide the student toward discovering a better solution themselves rather than directly stating a better solution. A particular focus was put on the ability of the students to explain in simple terms both what they had done and what the guard rail code provided by us did, as well as in writing understandably formatted and commented code. The eventual performance of the neural networks and execution time of the code, unless so poor as to call into question whether the task was actually solved, was not relevant for the grade. With respect to the gradually increasing difficulty of the programming tasks over the course of the semester, we applied increased leniency when grading the later tasks.

Our grading system followed a simple integer score between 0 (lowest mark) and 3 (highest mark) for each programming task, depending on the students' understanding of the core concepts related to the task, whether their code solutions solved it, and whether they needed significant assistance from the tutors.

Presentations were held at the start of a lesson and had the format of a classical presentation using slides and the blackboard, were around 15–30 minutes in length, and were followed by a plenary discussion in which the audience would ask the presenting students questions for clarifications. Additionally, the tutors would ask questions to cover nuances not explored by the presenting team, elaborate on inaccuracies or ambiguities in the presentation (if any), or gauge the level of understanding the presenters had of specific facts.

Similar to code submissions, an integer score between 0 and 3 was assigned depending on the quality of the presentations, the individual student's contribution, and their ability to answer questions by the other students or tutors.

The grading system at German universities features ten passing grades (1.0 being the best and 4.0 being the worst) and one failing grade (5.0). Thus, we first decided on a point threshold required for the worst passing grade (6 points) with every increment in grade requiring 1 additional point. The best possible grade (at a threshold of 15) was achievable without earning all possible (18) points.

4.3 Software and Data

In this section, we will briefly discuss the tools with which we taught the students to work.

4.3.1 Python

Python [58] is a high-level, general-purpose programming language based on C that has been under continuous development since 1989. It is particularly popular in the scientific community and among hobbyists for several reasons. Firstly, the language fully supports a number of different programming paradigms (such as object-oriented and functional programming). Secondly, the language is dynamically typed and uses a garbage collector for memory management, simplifying the process of creating programs in terms of difficulty and time spent developing. The language syntax has been explicitly streamlined for code readability. Finally, the language is also interpreted, significantly expediting the run and debug cycle, and sacrificing some performance for ease of use. Python's low bar of entry for beginners and its suitability for quickly prototyping code make it very popular among researchers. Additionally, the corresponding ecosystem of tutorials, standard and third-party libraries, and forums on which knowledge about the code base is shared is enormous, especially where ML and DL are concerned. In keeping with our premises from Sect. 2.1, we chose Python as the programming language on which to base our learning materials. Each lesson block comes with one or more collections of prepared code in the form of *IPython Notebooks*, in which code is segmented into individual cells, segments of code that can be executed and debugged independently of one another, a style of programming particularly suited for programming novices.

4.3.2 PyTorch

PyTorch [49] is a free, open-source software library based on Python and C++, developed for DL and Natural Language Processing (NLP). It interfaces high-level Python instructions with the CUDA API, allowing programmers to effectively utilize modern GPUs to speed up calculations. PyTorch is widely used, actively maintained, and (until the introduction of eager execution by its main alternative Tensorflow [67]) easier to debug for novices due to its default sequential execution.

4.3.3 Git

GitHub [20] is a Web-based platform for version control and collaboration that allows developers to easily collaborate on projects. Git is a distributed version control system that is used to manage projects and source code.

Git is an essential tool for computer science students to learn because it is used widely in the software development industry, and mastering Git is critical for collaboration on software projects. It allows developers to track changes in code, roll back to previous versions, and collaborate with other developers while maintaining a clear and organized code base.

4.3.4 The LiTS Dataset

For the sake of simplicity, we chose to base our course entirely on one dataset. The requirements we considered were a dataset of size small enough to be trainable in acceptable time and large enough not to cause overfitting to noise even if no or little data augmentation is applied, easy enough to quickly debug DL models and arrive at better solutions, and difficult enough that two models of different quality produce noticeably different results. Working with the data should be as uncomplicated as possible, which made datasets composed of raw Digital Imaging and Communications in Medicine (DICOM) files (a medical image format including both image and meta-information), or those with non-RGB, non-grayscale unattractive. Additionally, transformations on the dataset had to be intuitive – for example, random Gaussian noise is a realistic image augmentation for natural images (thermal noise in the camera) or computed tomography (CT) scans (thermal noise in the detector). However, the same is not quite true for magnetic resonance imaging (MRI) data, where realistic noise on an image is typically modeled as Rician noise [23]. Similarly, the values in a CT image correspond to a physical property of the scanned tissue, the attenuation of X-ray radiation coming from and being detected by the scanner (measured as an absolute quantity called Hounsfield Units, or HU), meaning they should in principle always be the same if the scanned object remains the same. In an MRI image, absolute values have less meaning and may vary from machine to machine.

Following the above reasoning, we used the LiTS (Liver Tumor Segmentation) dataset [8]. It consists of around 35,500 training images and 3000 validation and test images each. Every image is a 2-dimensional slice coming from 1 of the 201 individual 3-dimensional abdominal CT scans from patients with liver lesions. All images were converted to the PNG format and have 256×256 pixels with a single channel. A multitude of different models have been tested on this dataset and achieved a wide range of performances. On the GPUs described in Sect. 3, a single epoch of LiTS training can be completed in 1–2 minutes or less, depending on the specific model architecture, training setup, and caching. As training a model on LiTS typically takes 10 epochs or fewer, it allows students to evaluate any changes they make to their models in terms of performance and speed relatively quickly. Several typical LiTS images and their segmentations can be seen in Fig. 4.

For the AlexNet, ResNet, and ViT architectures, we framed the LiTS data as a classification problem with three classes: "no liver visible," "liver visible," and "liver visible, with lesion." For the U-Net architecture, we used the LiTS dataset with the same classes, but in the context of slice-wise semantic segmentation.

4.4 Topics and Tasks

In the following sections, we will cover the specific contents of each block of lessons. Each lesson block, except the first one, included a presentation. The guard

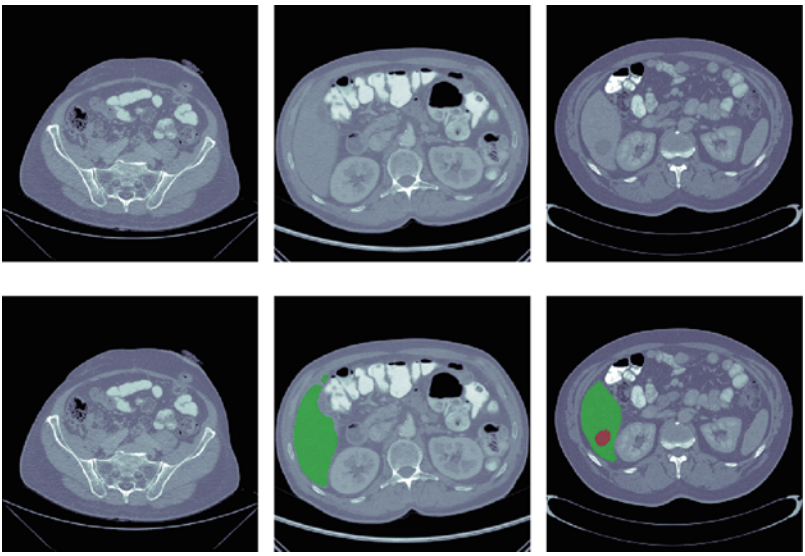


Fig. 4 Typical LiTS images. The images shown are two-dimensional slices taken from a three-dimensional scan. All images were contrast-enhanced. From left to right, they represent the classes 0 (no liver or lesion in the image), 1 (liver visible), and 2 (liver visible, lesion visible). In the second row, liver segmentations are marked in green, and lesion segmentations in red. The neural networks of the students must later identify every pixel in an image as belonging to one of these three classes

rail code, explanations, and sample solutions created by the student teams can be found at [29].

4.4.1 Introduction to Python

The first block was reserved for organizational work. We established the content of the lessons for the rest of the semester and laid out the grading system and learning objectives to the students. We also monitored the formation of the teams and introduced the students to the Coder platform through which the course's learning materials could be accessed.

The programming section of this block contained a tutorial covering the basics of programming with Python, starting with simple concepts that the students were already familiar with, such as simple calculus and linear algebra, declarations, and functions, and covering more advanced concepts specifically relevant to machine learning, such as working with vector- or matrix-type data arrays, error handling, and simple classes. The students were free to complete any or all sections of this tutorial according to their own evaluation of their skills under the premise that they would be familiar with the concepts introduced in the tutorial by the beginning of the next block. While there were not yet any graded tasks, the tutorial occasionally

contained small programming challenges, which tested the students' understanding of one particular concept.

Lastly, the students were introduced to backing up (or versioning) projects using GitHub. On the one hand, this would prevent data loss, even if the persistent storage behind one or more students' workspaces failed, and on the other hand, it introduced the students to best practices in coding.

4.4.2 Introduction to Deep Learning in PyTorch

The second lesson block formed the initial introduction to the topic of neural networks. A short presentation detailed the history of ML and the conceptualization of NN beginning with the earliest theoretical ideas [57] and culminating with the LeNet [37]. Additionally, it provided a concise explanation of the mathematical foundations of backpropagation [59] to those students who had no prior exposure to these methods. This presentation was held by us and served as an example of how to present, as well as providing the necessary foundation for later presentations.

The corresponding programming section introduced the students to the programming package PyTorch [49] and the basic elements needed to successfully train a NN in practice, such as:

- The dataset and dataloader classes (fetching the data)
- The NN model class and its forward pass (turning an image into a prediction)
- The loss function (asking, in layman's terms, "How wrong were my model's predictions?")
- The optimization of the model through backpropagation (asking "How would I have to adjust the model's parameters for it to be less wrong?")
- The validation of the model using some metric (asking "How good is my trained model?")

and how they are programmed using PyTorch.

The declared goal of this block was that the students would learn about each of these elements and build each of these elements on their own, producing one complete, end-to-end training and testing script. Even more so than for other blocks, the key ingredient for the best grade was not model performance or speed, but rather building a functional example of each element and being able to explain what it does and why it does it in the chosen way. We provided as guard rail code one possible variation of each of these elements, which the students were allowed to draw inspiration from.

4.5 Neural Network Implementations

As mentioned previously, we covered four milestone architectures from the field of image recognition during the seminar: AlexNet [35], ResNet [24], U-Net [56],

and the Vision Transformer [13]. These architectures were chosen due to their historical significance both in the field of computer vision in general and in medicine specifically.

For their presentations, the students were asked to cover the following core topics:

- What was the state of the art in image classification/segmentation before this milestone architecture?
- Describe and explain the most important components of the architecture. Which of these components were novel?
- Explain mathematically and in descriptive terms how the new operation works and why it is beneficial.
- Compare the performance of this architecture with those of its predecessors.

During the programming section, the students' task was to create the model class of the milestone architecture in PyTorch. Every subsequent architecture came with increasing complexity, slowly ramping up the difficulty as the students became more experienced:

- AlexNet only required the sequential application of pre-existing PyTorch modules, like the previous lesson block, and served to consolidate existing knowledge.
- ResNet required factoring out the basic building block of the ResNet into a separate class. The learning objective was to recognize this necessity, implement this subclass, and dynamically build the model from a number of these basic blocks.
- The U-Net lessons had multiple learning objectives. To build the U-Net, students had to understand the concepts of segmentation and the transposed convolution and had to apply the lessons learned during the ResNet implementation in a different context, again factoring out separate upsampling and downsampling blocks as subclasses. Additionally, the students were required to create the loss function for this task themselves.
- The implementation of the Vision Transformer model was the final task and again had multiple learning objectives. The students needed to understand both the concept and mathematics behind the attention mechanism [68], again factor it out as a separate class, and successfully implement it. Additionally, they had to internalize the concepts of embedding and the cls-token [68].

Every part of the training pipeline other than the model class (and the loss function in the case of the U-Net) was provided by us as guard rail code, although the students were encouraged to try and play around with various parameters or optimization of our code once the model class was functional. For each paper, the programming assignment was to approximate the original model described in that paper as closely as possible, occasionally with some decreases in overall parameter size. For the U-Net specifically, the assignment also asked for the implementation of a segmentation loss, incorporating both pixel-wise cross-entropy loss and a DICE-based loss. All lesson blocks concluded with the presentation of a sample solution and its performance by the same team that presented the architecture at its start.

5 The Seminar in Practice

In this section, we will discuss how the seminar panned out from an organizational point of view, and whether we can consider the learning objectives to have been reached. Given the grading system, we described in Sect. 4.2, this discussion will be qualitative in nature, although we will provide some quantitative context, where possible.

5.1 Introduction to Python

The organizational segment of the seminar and the subsequent introduction to Python proceeded according to the timetable. As there were no graded tasks and the learning objective was general familiarity with the basics of Python programming, there are no quantifiable indicators for the individual levels of success of the students. However, as the learning objectives of later topics were reached by all students, they necessarily had to have suitably familiarized themselves with the concepts in the introduction.

5.2 Introduction to Deep Learning with PyTorch

The introduction to DL in PyTorch proceeded with considerably more questions and challenges, particularly where multiple programming concepts had to be combined or where PyTorch-specific syntax was not known. We eventually decided to extend the lesson block by 1 week to allow the students more time to experiment and discuss questions with us. The graded programming task of this lesson block, one end-to-end training script where all necessary components were either PyTorch-native or programmed by the students, was satisfactorily completed by all students, and the students demonstrated sufficient conceptual understanding of all involved components to continue. The additional available time contributed significantly to this outcome.

5.3 Neural Network Implementations

The presentation, implementation, and discussion of the milestone architecture papers was conducted as described in Sects. 4.1 and 4.5. Although the students occasionally required tutor support for debugging, the frequency of such instances decreased over time, despite the increasing complexity of the architectures to implement. Simultaneously, the average severity and complexity of the issues brought

forth by the students increased over time – typical syntax errors were resolved increasingly independently by the students, leaving only difficult and unintuitive problems (e.g., relating to CPU-GPU communication or implicit operations that happen during class inheritance). This came in addition to students actively pointing out cases where divergences between the implementation details gathered from the respective papers and the guard rail code provided by us occurred, implying that the students' understanding of both Python and PyTorch in general and the subject matter specifically increased considerably.

By the end of the seminar, all programming tasks were solved by the majority of students, with points lost distributed across the semester, implying that the learning objectives were generally reached by all students. The implementation of the ViT architecture, for which we implicitly reduced the available time when extending the PyTorch tutorial block, was ultimately considered as extra credit (see Sect. 4.2). However, the group responsible for presenting the corresponding paper and constructing the sample solution completed these tasks even without making use of a potentially extended deadline, which suggests that by the end of the semester, this task would have been well within the capabilities of the other students to solve within the original timeframe, had the task remained mandatory. Overall, this is corroborated by the students themselves, who professed a marked increase in their programming abilities in Python and knowledge of DL (see Sect. 7). All students passed the seminar.

Additionally, the test-time performance of the student-implemented models and their choice of hyperparameters can serve as an indicator of their understanding of the subject matter. The U-Net sample solution, for example, achieved a test-time performance of 0.8268 DICE score on the lesion subclass and 0.9602 DICE score on the liver subclass (which would rank them at 25th out of 150 for lesions and tie them for 12th out of 66 for livers, respectively, on the CodaLab leaderboards [39]), even without the use of any data augmentation or advanced training strategies. Similarly, the sample solutions for the other architectures reliably achieved classification accuracies of 95% or above after only a few training epochs while training from scratch (without any additional pretraining). Where possible, the students trained their models with the hyperparameters of the original research papers, but every presenting group also experimented with other hyperparameters, finding combinations more suitable to training on the LiTS data, on their own initiative.

5.4 Estimated Time Expenditure

For a classical lecture style (without flipped classroom), lecture materials need to be prepared for each lesson in addition to the programming tasks. This can take on the order of days, depending on the complexity of the topic. In our flipped classroom approach, we successfully outsourced most of this time expenditure to the students, ostensibly with no reduction in the effectivity of the learning process, and tutors only

needed to briefly note down desired talking points and conclusions of the student presentations in advance.

The preparation of the learning materials was a one-time effort on the order of several weeks, with negligible maintenance (i.e., bug-fixing or refactors) once the materials were largely established. Both time expenditures are naturally eliminated if our published materials are reused, and only a quick familiarization on the order of a day is required for a ML professional to be able to help students with them.

While the time investment of our initial technical setup, several weeks in total, was significant compared to a custom solution based on local resources and unique student workspaces, we argue that it paid off in multiple ways. Firstly, being built on cloud computing resources, our platform can be reused by anyone, anywhere, at significantly reduced setup time, depending on familiarity with AWS and moving parts like Kubernetes. Secondly, we managed to avoid workspace-related issues entirely in practice, having been exclusively limited to either programming- or ML-related student questions.

The necessity of live maintenance of our platform was limited to several days across several months of use, similar to a custom solution and potentially even reduced by comparison, as the compute resources themselves never needed to be managed by us, except for provisioning.

Beyond accepting submissions, the necessary tutor-student interactions outside of seminar hours were surprisingly manageable and could be handled by a single tutor investing several hours per week. The simplicity of the debugging process during any of these instances can be attributed to the static elements (common access to workspaces, common environment, common guard rail code, etc.) of the learning materials introduced by our original design choices.

During the seminar itself, four (occasionally only three) tutors were more than sufficient to oversee the programming exercises and submissions of our ten students, each spending roughly half the time engaged actively helping a student. As a rough guideline, we expect that one tutor can comfortably handle two student teams of two or three people each. This number should be adjusted depending on the prior programming experience of the students and time available for the Introduction lessons – highly inexperienced students will require frequent supervision while those with significant programming experience may be able to spend multiple sessions without any supervision.

6 Discussion

According to our own assessment and student feedback (see Sect. 7), the seminar was very successful and its learning objectives were achieved by all students. The fact that half of the students decided to pursue master theses in machine learning-related topics under the supervision of our respective faculties corroborates this assessment.

We observed that the steepest learning part of the learning curve occurs when the training loop inherent to PyTorch is first introduced to the students, as it requires both becoming familiar with PyTorch concepts and applying every concept learned earlier at once. There are two lessons to draw from this. On the one hand, the ad hoc decision to allocate extra time to this lesson block was justified. This extra time can be drawn either from the Python tutorial at the start, simply requiring a better understanding of Python as a prerequisite to the seminar, or (as in our case) it can be gained by reducing the available time on the last lesson block, in which case any points a student can receive from it should strictly be considered extra credit and not mandatory. In general, the dynamic adjustment of lesson blocks during the seminar is necessary for an optimal learning experience.

The quality of the individual student presentations and the detailed content of the questions asked both by the students and occasionally the presenters confirmed to us that letting the students introduce one another to the content significantly increased their investment in the process and helped them accurately express themselves in the technical jargon and more precisely specify any questions that they could not research themselves, without reducing the quality or level of detail within the presented materials. Additionally, while we believe that holding the seminar in a hybrid or fully remote fashion is possible, both our observations and student feedback indicate that attending the seminar in person (which students almost exclusively did) improves communication both between tutors and students and within the student teams significantly.

From conversations with the students, it has become apparent that the level of engagement with the subject matter was enabled by the ease with which the learning materials could be accessed and worked on. The flipped classroom concept was also well-received and working in groups in particular was strongly appreciated by students. The majority of time "lost" to matters not directly related to the then-current task was caused by what few system outages we experienced due to maintenance and due to the few instances in which we introduced small fixes or refactors to the guard rail code the students were using. Consequently, the minimization of logistical frustration and maximization of system robustness were heavily implied to be useful design principles. Overall, we conclude that the flipped classroom concept was helpful to both tutors and students and that this would be the case almost irrespective of the precise topic, so long as the students can be expected to understand the materials they present.

From the tutor perspective, we also considered the platform to have played a significant part in the success of the seminar. The ability to remotely view and execute the code of students was consistently used during every class and between classes. In many cases, it enabled the normal participation of students that would otherwise have been absent, as well as the submission and discussion of solutions outside of the seminar, where needed.

Previous observations on the efficacy of a flipped classroom in STEM fields [27] suggest that a majority of practical implementations resulted in both positive feedback and improved student performance. Students typically reported being more engaged with the subject matter and, unless inundated with the learning

materials, reported enjoying the cooperative learning process. Students' assessments of their self-efficacy, however, were reportedly mixed. During the seminar, we largely made the same observations. We further observed that reported self-efficacy appeared positively correlated with student performance, indicating that the success of the flipped classroom relies on a good foundation among students. Tutors must ensure that all students possess or acquire this foundation so that the students can work on their own. These conclusion are in line with those gathered in [27].

The fact that submissions were made individually, a time-consuming process and that the number of available topics was limited by the time available to actually complete a programming task related to that topic, means that with an increasing number of students, appropriately dividing the presentations among students becomes a nontrivial process. This comes in addition to the issue of a limited number of tutors helping individual students or teams debug their code (with which the tutor themselves is only partially familiar as well), a problem that may quickly outscale the capacity of tutors as the number of students increases. This is something to keep in mind when adapting our seminar's strategies for larger audiences.

With regard to the technical design, we observed that it was feasible to implement a cloud-hosted remote development platform for enriching courses with GPU-focused practical sessions. While we noticed that during demand spikes, such as at the start of each lesson, the resource provisioning could slow down, this rarely caused issues in practice. For seminars of a bigger scale, this potential limitation can be addressed with proprietary solutions such as Amazon SageMaker [60], although this would negate our vendor neutrality goal. In the future, we plan to move our platform on-premises.

Despite the evident completion of the learning objectives and the general student satisfaction, we must be careful not to overstate the relevance of the seminar's design principles.

Firstly, at a sample size of ten students, any conclusion we draw for the success of the seminar comes with the qualifier that our experience may not generalize to larger groups of students or students of different skill levels. While most of the seminar was designed to account for increasing numbers of students (e.g., via the centrally managed containerized environment or ability to submit solutions outside of class hours), it is possible that those elements of the seminar that are not as scalable, such as the number of available tutors and their free time, can become a limitation. In practice, however, the typically available three to four tutors were more than sufficient for ten students and the seminar could easily have been run with more students. This limitation can also be addressed by requiring working in teams at all times (which we highly encouraged but never strictly enforced).

The majority of students in our seminar (8 out of 10) had heard a theory lecture breaking down the mathematical foundations of ML during the previous semester and most students entered the seminar with some – and a few with significant – prior experience in programming (see Appendix), which means that at least a baseline understanding of the seminar's subject matter was already present among the students prior to taking the seminar. While this does not undermine the success

of the seminar in itself (the students had not previously interacted with ML in medicine or in practice), it implies that the seminar should suggest at least a passing familiarity with the mathematical foundation of ML and/or some programming skill in Python as a prerequisite for taking the seminar. We consider this particularly true where the flipped classroom approach we recommend is pursued, with literature being described as divided on whether a flipped classroom approach is suitable for introductory or foundation courses [27].

7 Conclusions and Outlook

We have designed a practical seminar on ML in medicine for university students with no or only some prior experience in ML and programming, based on a set of generalizable design principles. All learning materials were made available through an interactive cloud platform purpose-built for this seminar, through which computational resources were made available. The seminar covered an introduction to the programming language Python, the ML library PyTorch, and practical sessions reimplementing the architectures of the milestone ML models AlexNet, ResNet, U-Net, and ViT and evaluating them on the medical imaging challenge dataset LiTS. The technical realization of the seminar flexibly allowed for an in-person, hybrid, or remote lesson style. Furthermore, it relied entirely on a small number of highly adaptable scripts, and a single Web page, and otherwise only utilized publicly available (and in many cases free) resources. All of these scripts, as well as our learning materials and the students' sample solutions, were made publicly available.

Based on our own observations, the successful completion of all five programming tasks' learning objectives, and the students' personal feedback on the seminar and their active participation in it, we conclude that the seminar was a resounding success and that the learning materials and cloud platform we provide will help teachers and students integrate ML into the routines of medical research, education, and practice. In the future, we will continue to update and refine both the technical implementation of the cloud platform and the content of the learning materials.

Beyond the course-specific lessons, what else can we take away from this? For one, we believe that our experience demonstrates that practical machine learning is a surprisingly approachable topic. Further, we believe that even advanced ML topics could be taught earlier than during the master's degree, and even in fields where the average programming experience is typically low, such as medicine (where machine learning is already extremely relevant [14]). Given the sheer breadth of research disciplines in medicine that can profit from ML and the large fraction of them that have seen barely any influence from DL at the same time [14], the possibilities created by teaching DL to healthcare professionals involved in research can hardly be overstated.

Similarly, we believe that AI has a place in the educational sector, both as a tool and a topic. The majority of US secondary schools have already added computer science to the curricula of students (see, e.g., [55]) and, in our opinion, it is not unrealistic to imagine AI and ML becoming a limited part of these lessons as well. According to multiple sources (e.g., [17, 50, 51]), the notion that such a development is desirable is shared by many educators and experts in the field.

In a broader context, the past few years have seen DL research experience an incredible growth [72], both in the field itself and even in a variety of disciplines traditionally considered unrelated, from the creation of art or poems, personalized browsing or shopping experiences, over self-driving cars and robots, to the incidental detection of cancerous growths and advanced clinical tools. With further growth on the horizon, this trend will likely see DL become an inseparable part of more and more professions, both in academia and beyond. We posit that lessons on DL can and should be taught to students across more age groups and subjects than it is now. As AI continues to conquer the world and make the previously impossible possible, the broad perspective on the possibilities of its use (and misuse), which we can create in this manner, will be invaluable – and now is the time to make sure that it exists.

Acknowledgments The hosting costs of the computational hardware were fully covered by Amazon by means of credits issued for research purpose.

Appendix

About Us

The Institute for AI in Medicine (IKIM) is a research institution at the University Hospital of Essen, Germany, and the University Duisburg-Essen (Uni-DUE). Founded in 2019, the institute employs researchers, engineers, and healthcare professionals across multiple separate research groups covering both theoretical and practical research in data science, ML, explainable and trustworthy AI, and clinical applications in cancer and sepsis research, radiology, and the development of clinical tools.

Our research group, Medical Machine Learning, features a diverse staff from a variety of backgrounds including both computer science and medicine.

Closely tied to the institute is the University of Duisburg-Essen, which was established in 2003. Its Faculty of Mathematics, which offered the seminar that we discuss in the paper, has been ranked among the top ten departments of mathematics in Germany, according to the Academic Ranking of World Universities (Shanghai Ranking) 2022. Starting in 2019 the chair of Numerical Mathematics at the faculty has offered lectures and seminars in ML.

Student Questionnaire

After the seminar concluded, we asked students to fill out a small questionnaire regarding the seminar, including informal feedback which we already discussed. Some of the results from that questionnaire are shown in this section. Note that with only nine questionnaires, any conclusions taken from these results should be taken with a grain of salt. These results are shown in Figs. 5, 6, 7, 8, 9, and 10.

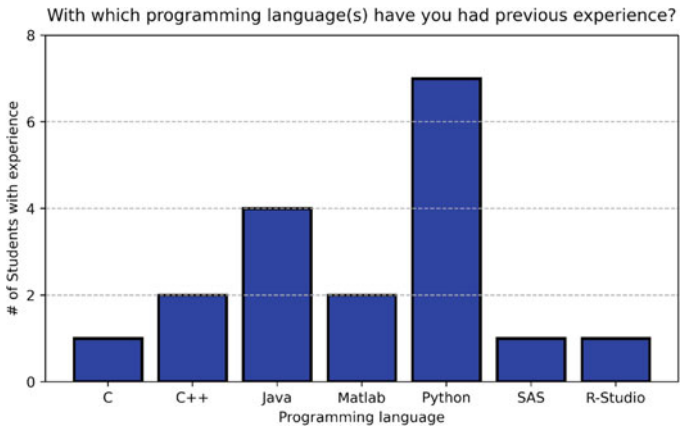


Fig. 5 Previous programming experience of participating students

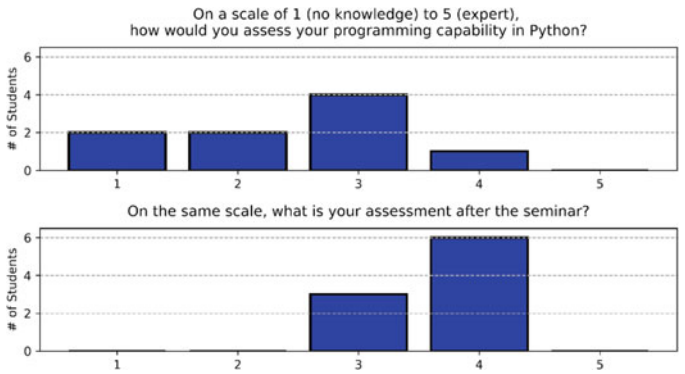


Fig. 6 Self-assessment of programming ability in Python before and after the seminar

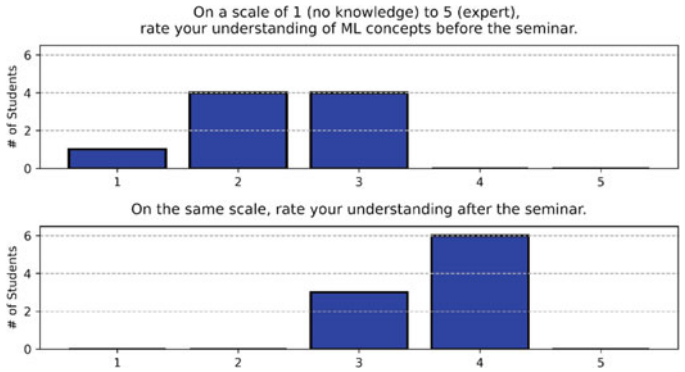


Fig. 7 Self-assessment of understanding of ML concepts before and after the seminar

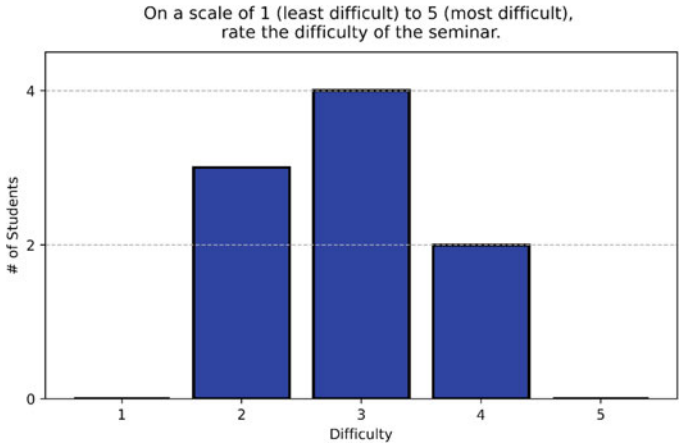


Fig. 8 Subjective assessment of the difficulty of the seminar

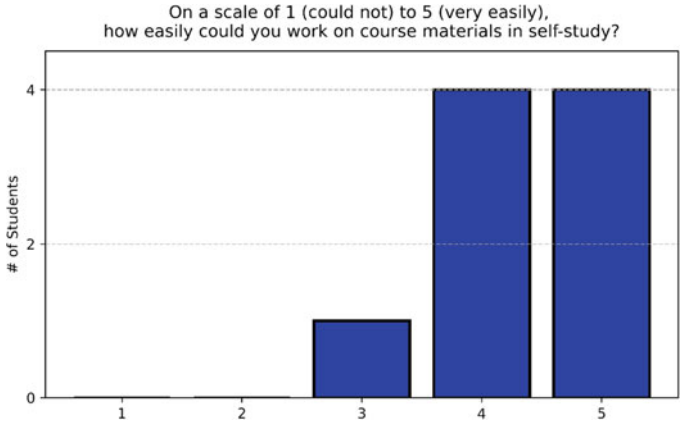


Fig. 9 Self-assessment of ability to work on course materials self-sufficiently

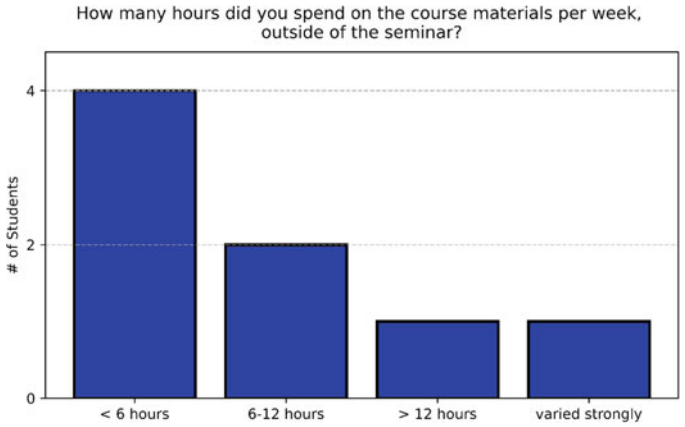


Fig. 10 Time spent on course materials per week

References

1. Amazon Elastic File System. https://aws.amazon.com/de/efs/
2. Amazon Elastic Kubernetes Service. https://aws.amazon.com/de/eks/
3. Amazon Relational Database Service. https://aws.amazon.com/de/rds/
4. Bahnsen, F.H., Fey, G.: Yaps-your open examination system for activating and empowering students. In: 2021 16th International Conference on Computer Science & Education (ICCSE), pp. 98–103. IEEE, Piscataway (2021)
5. Bahnsen, F.H., Sakhri, M., Dillenberger, J., Sitzmann, D.: Skalierbares e-assessment mit docker und ansible. In: E-Prüfungs-Symposium (ePS) (2022)
6. Bandukwala, A., Carberry, K., Entwistle, J.A.: Coder - your self-hosted remote development platform. https://coder.com
7. Biggs, J.: Enhancing teaching through constructive alignment. Higher Educ. **32**(3), 347–364 (1996)
8. Bilic, P., Christ, P., Li, H.B., Vorontsov, E., Ben-Cohen, A., Kaissis, G., Szeskin, A., Jacobs, C., Mamani, G.E.H., Chartrand, G., Lohöfer, F., Holch, J.W., Sommer, W., Hofmann, F., Hostettler, A., Lev-Cohain, N., Drozdzal, M., Amitai, M.M., Vivanti, R., Sosna, J., Ezhov, I., Sekuboyina, A., Navarro, F., Kofler, F., Paetzold, J.C., Shit, S., Hu, X., Lipková, J., Rempfler, M., Piraud, M., Kirschke, J., Wiestler, B., Zhang, Z., Hülsemeyer, C., Beetz, M., Ettlinger, F., Antonelli, M., Bae, W., Bellver, M., Bi, L., Chen, H., Chlebus, G., Dam, E.B., Dou, Q., Fu, C.W., Georgescu, B., Giró-i Nieto, X., Gruen, F., Han, X., Heng, P.A., Hesser, J., Moltz, J.H., Igel, C., Isensee, F., Jäger, P., Jia, F., Kaluva, K.C., Khened, M., Kim, I., Kim, J.H., Kim, S., Kohl, S., Konopczynski, T., Kori, A., Krishnamurthi, G., Li, F., Li, H., Li, J., Li, X., Lowengrub, J., Ma, J., Maier-Hein, K., Maninis, K.K., Meine, H., Merhof, D., Pai, A., Perslev, M., Petersen, J., Pont-Tuset, J., Qi, J., Qi, X., Rippel, O., Roth, K., Sarasua, I., Schenk, A., Shen, Z., Torres, J., Wachinger, C., Wang, C., Weninger, L., Wu, J., Xu, D., Yang, X., Yu, S.C.H., Yuan, Y., Yue, M., Zhang, L., Cardoso, J., Bakas, S., Braren, R., Heinemann, V., Pal, C., Tang, A., Kadoury, S., Soler, L., van Ginneken, B., Greenspan, H., Joskowicz, L., Menze, B.: The liver tumor segmentation benchmark (LiTS). Med. Image Anal. **84**, 102680 (2023). https://doi.org/10.1016/j.media.2022.102680.
9. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324.

10. Cockburn, A., Williams, L.: The Costs and Benefits of Pair Programming, pp. 223–243. Addison-Wesley Longman Publishing, USA (2001)
11. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018.
12. Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: towards explaining agile software development. J. Syst. Softw. **85**(6), 1213–1221 (2012). https://doi.org/10.1016/j.jss.2012.02.033
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16×16 words: transformers for image recognition at scale. Tech. Rep. arXiv:2010.11929, arXiv (2021). http://arxiv.org/abs/2010.11929
14. Egger, J., Gsaxner, C., Pepe, A., Pomykala, K.L., Jonske, F., Kurz, M., Li, J., Kleesiek, J.: Medical deep learning—a systematic meta-review. Comput. Methods Progr. Biomed. **221**, 106874 (2022). https://doi.org/10.1016/j.cmpb.2022.106874
15. Egger, J., Wild, D., Weber, M., Bedoya, C.A.R., Karner, F., Prutsch, A., Schmied, M., Dionysio, C., Krobath, D., Jin, Y., et al.: Studierfenster: an open science cloud-based medical imaging analysis platform. J. Digital Imaging **35**(2), 340–355 (2022)
16. Ester, O., Hörst, F., Seibold, C., Keyl, J., Ting, S., Vasileiadis, N., Schmitz, J., Ivanyi, P., Grünwald, V., Bräsen, J.H., Egger, J., Kleesiek, J.: Valuing vicinity: memory attention framework for context-based semantic segmentation in histopathology (2022). https://doi.org/10.48550/arXiv.2210.11822
17. Evangelista, I., Blesio, G., Benatti, E.: Why are we not teaching machine learning at high school? A proposal. In: 2018 World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC), pp. 1–6. IEEE, Albuquerque (2018). https://doi.org/10.1109/WEEF-GEDC.2018.8629750.
18. Freeman, S., Eddy, S.L., McDonough, M., Smith, M.K., Okoroafor, N., Jordt, H., Wenderoth, M.P.: Active learning increases student performance in science, engineering, and mathematics. Proc. Natl. Acad. Sci. U.S.A. **111**(23), 8410–8415 (2014). https://doi.org/10.1073/pnas.1319030111
19. Fukushima, K.: Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybern. **36**(4), 193–202 (1980). https://doi.org/10.1007/BF00344251
20. github: GitHub (2020). https://github.com/
21. GitHub Codespaces. https://github.com/features/codespaces
22. GitLab Remote Development. https://docs.gitlab.com/ee/user/project/remote_development/
23. Gudbjartsson, H., Patz, S.: The Rician distribution of noisy MRI data. Magn. Reson. Med. **34**(6), 910–914 (1995). https://doi.org/10.1002/mrm.1910340618
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, Las Vegas, NV, USA (2016). https://doi.org/10.1109/CVPR.2016.90
25. Heiliger, L., Marinov, Z., Hasin, M., Ferreira, A., Fragemann, J., Pomykala, K., Murray, J., Kersting, D., Alves, V., Stiefelhagen, R., Egger, J., Kleesiek, J.: Autopet challenge: combining nn-unet with swin unetr augmented by maximum intensity projection classifier (2022)
26. Hosch, R., Kattner, S., Berger, M.M., Brenner, T., Haubold, J., Kleesiek, J., Koitka, S., Kroll, L., Kureishi, A., Flaschel, N., Nensa, F.: Biomarkers extracted by fully automated body composition analysis from chest CT correlate with SARS-COV-2 outcome severity. Sci. Rep. **12**(1), 16411 (2022). https://doi.org/10.1038/s41598-022-20419-w.
27. Huber, E., Werner, A.: A review of the literature on flipping the stem classroom: preliminary findings. In: 33rd International Conference of Innovation, Practice and Research in the Use of Educational Technologies in Tertiary Education-ASCILITE 2016-Show Me the Learning (2016)
28. Isensee, F., Jaeger, P.F., Kohl, S.A.A., Petersen, J., Maier-Hein, K.H.: nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nat Methods **18**(2), 203–211 (2021). https://doi.org/10.1038/s41592-020-01008-z

29. Jonske, F., Dada, A., Fragemann, J., Kleesiek, J., Kraus, J., Lymbery, M., Osthues, K.: Teaching Machine Learning in Medicine, Learning Materials (2023).https://github.com/TIO-IKIM/ML_in_Medicine_Interactive_Seminar.
30. Jonske, F., Dederichs, M., Kim, M.S., Keyl, J., Egger, J., Umutlu, L., Forsting, M., Nensa, F., Kleesiek, J.: Deep learning–driven classification of external DICOM studies for PACS archiving. Eur. Radiol. **32**(12), 8769–8776 (2022). https://doi.org/10.1007/s00330-022-08926-w
31. Kaul, V., Enslin, S., Gross, S.A.: History of artificial intelligence in medicine. Gastrointestinal Endosc. **92**(4), 807–812 (2020). https://doi.org/10.1016/j.gie.2020.06.040
32. Kim, M., Seifert, R., Fragemann, J., Kersting, D., Murray, J., Jonske, F., Pomykala, K.L., Egger, J., Fendler, W.P., Herrmann, K., Kleesiek, J.: Evaluation of thresholding methods for the quantification of ^{68}Ga-PSMA-11 pet molecular tumor volume and their effect on survival prediction in patients with advanced prostate cancer undergoing ^{177}Lu-PSMA-617 radioligand therapy. Eur. J. Nucl. Med. Mol. Imaging 50(7), 2196–2209 (2023)
33. Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C.: Jupyter notebooks – a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (eds.) Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87 – 90. IOS Press (2016)
34. Koitka, S., Gudlin, P., Theysohn, J.M., Oezcelik, A., Hoyer, D.P., Dayangac, M., Hosch, R., Haubold, J., Flaschel, N., Nensa, F., Malamutmann, E.: Fully automated preoperative liver volumetry incorporating the anatomical location of the central hepatic vein. Sci. Rep. **12**(1) (2022). https://doi.org/10.1038/s41598-022-20778-4
35. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J., Bottou, L., Weinberger, K.Q. (eds.) Advances in neural information processing systems, vol. 25. Curran Associates (2012). https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
36. Kubernetes Cluster Autoscaler. https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler
37. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
38. Li, J., Pepe, A., Gsaxner, C., Egger, J.: An online platform for automatic skull defect restoration and cranial implant design (2020). arXiv preprint arXiv:2006.00980
39. LiTS - Liver Tumor Segmentation Challenge. https://competitions.codalab.org/competitions/17094#results
40. Lübke, O., Fuger, K., Bahnsen, F.H., Billerbeck, K., Schupp, S.: How to derive an electronic functional programming exam from a paper exam with proofs and programming tasks. In: Trends in Functional Programming in Education (TFPIE) (2023)
41. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent. In: Solla, S., Leen, T., Müller, K. (eds.) Advances in Neural Information Processing Systems, vol. 12. MIT Press, Cambridge (1999). https://proceedings.neurips.cc/paper_files/paper/1999/file/96a93ba89a5b5c6c226e49b88973f46e-Paper.pdf
42. Merkel, D.: Docker: lightweight Linux containers for consistent development and deployment. Linux J. **2014**(239), 2 (2014)
43. Microsoft Dev Box.https://learn.microsoft.com/en-us/azure/dev-box/overview-what-is-microsoft-dev-box
44. Nasca, E.: Teaching Machine Learning in Medicine, Technical Specifications of the Interactive Cloud Platform (2023). https://github.com/TIO-IKIM/coder-aws
45. Nouri, J.: The flipped classroom: for active, effective and increased learning—especially for low achievers. Int. J. Educ. Technol. Higher Educ. **13**, 1–10 (2016)
46. NVIDIA GPU Operator. https://github.com/NVIDIA/gpu-operator
47. O'Shea, K., Nash, R.: An Introduction to Convolutional Neural Networks. Tech. Rep. arXiv:1511.08458, arXiv (2015). http://arxiv.org/abs/1511.08458

48. Overview of Amazon Web Services - AWS Whitepaper (2022). https://docs.aws.amazon.com/pdfs/whitepapers/latest/aws-overview/aws-overview.pdf?did=wp_card&trk=wp_card
49. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., Alché-Buc, F.d., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf
50. Pedró, F., Subosa, M., Rivas, A., Valverde, P.: Artificial intelligence in education: challenges and opportunities for sustainable development (2019)
51. Polak, S., Schiavo, G., Zancanaro, M.: Teachers' perspective on artificial intelligence education: an initial investigation. In: CHI Conference on Human Factors in Computing Systems Extended Abstracts, pp. 1–7. ACM, New Orleans (2022). https://doi.org/doi/10.1145/3491101.3519866
52. Prutsch, A., Pepe, A., Egger, J.: Design and development of a web-based tool for inpainting of dissected aortae in angiography images (2020). arXiv preprint arXiv:2005.02760
53. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D.Y., Bagul, A., Langlotz, C.P., Shpanskaya, K.S., Lungren, M.P., Ng, A.Y.: Chexnet: radiologist-level pneumonia detection on chest x-rays with deep learning (2017). http://arxiv.org/abs/1711.05225
54. RedHat OpenShift Dev Spaces. https://developers.redhat.com/products/openshift-dev-spaces/overview
55. Roberts, S., Glennon, M., Weissman, H., Fletcher, C., Dunton, H., Baskin, J., Mak, J.: 2022 State of Computer Science Education. https://advocacy.code.org/stateofcs
56. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
57. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychol. Rev. **65**(6), 386–408 (1958). https://doi.org/10.1037/h0042519
58. Rossum, G.v., Drake, F.L.: The Python language reference, release 3.0.1 [repr.] edn. No. Pt. 2 in Python documentation manual/Guido van Rossum; Fred L. Drake [ed.]. Python Software Foundation, Hampton (2010)
59. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986). https://doi.org/10.1038/323533a0
60. SageMaker Studio Administration Best Practices - AWS Whitepaper. https://docs.aws.amazon.com/whitepapers/latest/sagemaker-studio-admin-best-practices/sagemaker-studio-admin-best-practices.html?did=wp_card&trk=wp_card
61. Salleh, N., Mendes, E., Grundy, J.: Empirical studies of pair programming for CS/SE teaching in higher education: a systematic literature review. IIEEE Trans. Software Eng. **37**(4), 509–525 (2011). https://doi.org/10.1109/TSE.2010.59
62. Scherer, J., Nolden, M., Kleesiek, J., Metzger, J., Kades, K., Schneider, V., Bach, M., Sedlaczek, O., Bucher, A.M., Vogl, T.J., Grúnwald, F., Kühn, J.P., Hoffmann, R.T., Kotzerke, J., Bethge, O., Schimmöller, L., Antoch, G., Müller, H.W., Daul, A., Nikolaou, K., la Fougère, C., Kunz, W.G., Ingrisch, M., Schachtner, B., Ricke, J., Bartenstein, P., Nensa, F., Radbruch, A., Umutlu, L., Forsting, M., Seifert, R., Herrmann, K., Mayer, P., Kauczor, H.U., Penzkofer, T., Hamm, B., Brenner, W., Kloeckner, R., Düber, C., Schreckenberger, M., Braren, R., Kaissis, G., Makowski, M., Eiber, M., Gafita, A., Trager, R., Weber, W.A., Neubauer, J., Reisert, M., Bock, M., Bamberg, F., Hennig, J., Meyer, P.T., Ruf, J., Haberkorn, U., Schoenberg, S.O., Kuder, T., Neher, P., Floca, R., Schlemmer, H.P., Maier-Hein, K.: Joint imaging platform for federated clinical data analytics. JCO Clin. Cancer Inform. (4), 1027–1038 (2020). https://doi.org/10.1200/CCI.20.00045

63. Seibold, C., Fink, M.A., Goos, C., Kauczor, H.U., Schlemmer, H.P., Stiefelhagen, R., Kleesiek, J.: Prediction of low-keV monochromatic images from polyenergetic ct scans for improved automatic detection of pulmonary embolism. In: 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), pp. 1017–1020 (2021). https://doi.org/10.1109/ISBI48211.2021.9433966
64. Strack, C., Pomykala, K.L., Schlemmer, H.P., Egger, J., Kleesiek, J.: 'a net for everyone': fully personalized and unsupervised neural networks trained with longitudinal data from a single patient (2022). https://doi.org/10.48550/arXiv.2210.14228
65. Sulmont, E., Patitsas, E., Cooperstock, J.R.: Can you teach me to machine learn? In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, pp. 948–954. ACM, Minneapolis (2019). https://doi.org/10.1145/3287324.3287392
66. Turing, A.M.: Computing machinery and intelligence. Mind **59**(236), 433–460 (1950). http://www.jstor.org/stable/2251299
67. TensorFlow Developers: TensorFlow (2023). https://doi.org/10.5281/ZENODO.4724125
68. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 30. Curran Associates (2017). https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
69. Visual Studio Code. https://github.com/microsoft/vscode
70. Visual Studio Code Remote Development. https://code.visualstudio.com/docs/remote/remote-overview
71. Wu, N., Phang, J., Park, J., Shen, Y., Huang, Z., Zorin, M., Jastrzebski, S., Févry, T., Katsnelson, J., Kim, E., Wolfson, S., Parikh, U., Gaddam, S., Lin, L.L.Y., Ho, K., Weinstein, J.D., Reig, B., Gao, Y., Toth, H., Pysarenko, K., Lewin, A., Lee, J., Airola, K., Mema, E., Chung, S., Hwang, E., Samreen, N., Kim, S.G., Heacock, L., Moy, L., Cho, K., Geras, K.J.: Deep neural networks improve radiologists' performance in breast cancer screening. IEEE Transa. Med. Imaging **39**(4), 1184–1194 (2020). https://doi.org/10.1109/TMI.2019.2945514
72. Zhang, D., Maslej, N., Brynjolfsson, E., Etchemendy, J., Lyons, T., Manyika, J., Ngo, H., Niebles, J.C., Sellitto, M., Sakhaee, E., Shoham, Y., Clark, J., Perrault, R.: The AI Index 2022 Annual Report. Tech. Rep. arXiv:2205.03468, arXiv (2022). http://arxiv.org/abs/2205.03468. ArXiv:2205.03468 [cs] type: article

Part III
Artificial Intelligence for Anomaly Detection

Machine Learning and Anomaly Detection for an Automated Monitoring of Log Data

Simone Falzone, Gabriele Gühring, and Benjamin Jung

1 Introduction

Anomaly detection deals with finding patterns in data sets and recognizing anomalies by deviations from these patterns, i.e., [1, 5, 33]. Methods for anomaly detection are used in many safety-critical areas because anomalies indicate abnormal operating conditions, e.g., a rotational error in an aircraft engine [16] or potential health complications when monitoring the heart rate of patients [32]. Algorithms for anomaly detection range from various conventional methods, i.e., statistical procedures or clustering methods as in [5] to deep learning as in [30], where an auto-encoder is presented for anomaly detection. A direct application of anomaly detection methods can be found in the monitoring of computer network log files to detect if the network traffic pattern is changing significantly; see [5, 14, 21–23], or [31]. Usually the running state of a network system is recorded in log files. The analysis of log files might be used for debugging [14], for fault detection, but also for program verification or performance prediction [8] or even for intrusion detection; see, for example, [21] and [12].

Since modern computer systems generate an ever-increasing number of log files and the corresponding log entries are generated at a fast rate, a human supervision of them becomes almost impossible or at least is very expensive; see also [22] or [31].

S. Falzone · B. Jung
AEB SE, Stuttgart, Germany
e-mail: simone.falzone@aeb.com; benjamin.jung@aeb.com

G. Gühring (✉)
Esslingen University of Applied Sciences, Esslingen, Germany
e-mail: gabriele.guehring@hs-esslingen.de

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_13

Therefore, over the last 20 years (see [34]), different methods for automatic anomaly detection in system logs have been researched. There exist supervised anomaly detection algorithms such as support vector machines [26], logistic regression [3], decision trees [7], or also artificial neural networks; see [4] or [21]. And there are unsupervised methods. One of them is principal component analysis (PCA), which is already used in [37] for log anomaly detection. Other unsupervised algorithms use clustering methods as described in [22] or deep learning methods as described in the surveys of [21] and [23]. Supervised algorithms have the disadvantage that they are trained on log files containing non-anomalous as well as anomalous entries. Therefore there are only a few labeled data sets, i.e., the Hadoop Distributed File System (HDFS) [37] or the BlueGene/L Supercomputer System (BGL) [12], which might be used to evaluate the algorithms. It is unclear if the results can be transferred to different log file systems. Therefore, we here present the three commonly known unsupervised algorithms PCA, [37], the clustering algorithm *LogCluster* [27], and the deep learning algorithm *DeepLog* first established in [11] for anomaly detection in log files. These three algorithms are often used as benchmark algorithms as stated in [23]. Different to other evaluations, we evaluate them on three different types of log files, which are all preprocessed in a different way. We show that whether they are successful or not depends on the data set that is examined.

All of the three algorithms rely on a log parser, i.e., *Drain* presented in [15] or *Spell* [10], when preprocessing the log data. Log parsing tries to extract event templates from raw log entries. The generated event templates play a major role when detecting anomalies in log data [36], and the noise from log parsing errors has impact on the performance of anomaly detection models [25]. However, some of the anomaly detection algorithms are better in dealing with the noise as others and some of the log files produce more noise than the others. Therefore, in the following sections we show that:

- structuring the log data entries in advance helps to detect anomalies in their structure
- algorithms, which are able to manage a large amount of different event templates, such as *LogCluster*, usually lead to a smaller rate of false-positive anomalies
- PCA and the clustering method *LogCluster* outperform on structured authentication log data even the results obtained with deep learning methods on HDFS.

While a lot of literature, i.e., [21, 38], or [31], is optimizing the algorithms, we show that the success of the algorithm is dependent on the form of the log entries as well as the preprocessing steps performed on the log entries. This way, we apply the three benchmark algorithms first on common semi-structured log data as in Fig. 1, then exception fingerprints as in Fig. 6 are used as preprocessed log entries, and lastly, we apply the algorithms to well-structured authentication logs.

2 Methods for Anomaly Detection in Log Data

Before giving a rough explanation of the three benchmark algorithms PCA in Sect. 2.3, the clustering method *LogCluster* in Sect. 2.4, and the deep learning algorithm *DeepLog* in Sect. 2.5, we present an overview on the phases of anomaly detection in log data. Not all of them are needed for the algorithms we present thereafter. Feature extraction [15] will only be used for PCA and the clustering algorithm *LogCluster* [27] but not for the deep learning algorithm *DeepLog*. The deep learning model *DeepLog* is directly trained on the sequences of the log keys. We refer to Figs. 14, 15, and 16 for a general overview of the steps needed for each algorithm. However, for all the methods that are dealt with in the following, usually a log parsing must be carried out at the beginning. The goal of log parsing is to split each log message into a constant and a variable part. By parsing the log data, each log entry is assigned a log template and a log key [24].

2.1 Log Parsing

A common log parser is *Drain*, a fixed **d**epth t**r**ee-b**a**sed onl**i**ne log parsi**n**g method, as first described in [15]. It can parse log entries in a streaming manner. *Drain* automatically extracts log templates from raw log messages and split them into disjoint log groups. According to [15], it employs a parse tree with fixed depth to guide the log group search process, which effectively avoids constructing a very deep and unbalanced tree. As described in [40], a common log statement consists of a header and a log message as in Fig. 1. The logging statement from Fig. 1 shows that only the log message is freely determined by software developers. The meta information such as the verbosity level or the time stamps are automatically added by the logging framework. Since this meta information is added automatically by the logging framework, the structure is always the same, which allows to easily extract this information. With the remaining free text part in the log message, it is much more difficult, since there is a constant and a variable part that can vary greatly; see [40]. The constant part consists of tokens that describe an operation within the system, such as `Received block <*> of size <*> from <*>` as in [15]. In between, the variable part of a log message is replaced with placeholders `<*>`. The result is an event template; see Fig. 1. For each event template, a unique event template ID, also called log key, i.e., E_0, E_1, E_2, E_3, ..., is assigned. Typical log parsers consider log parsing as a clustering problem and group each log message with the same event template in a log group; see [15]. Usually, when parsing log entries, regular expressions were used. But with modern computer systems, this is for various reasons, no longer practical. One reason is that the number of logs is increasing more and more, making a manual method unacceptable. Furthermore, the cycles in which modern computer systems are updated are becoming shorter, which is why the log entries also change or new ones are added. So, the developers

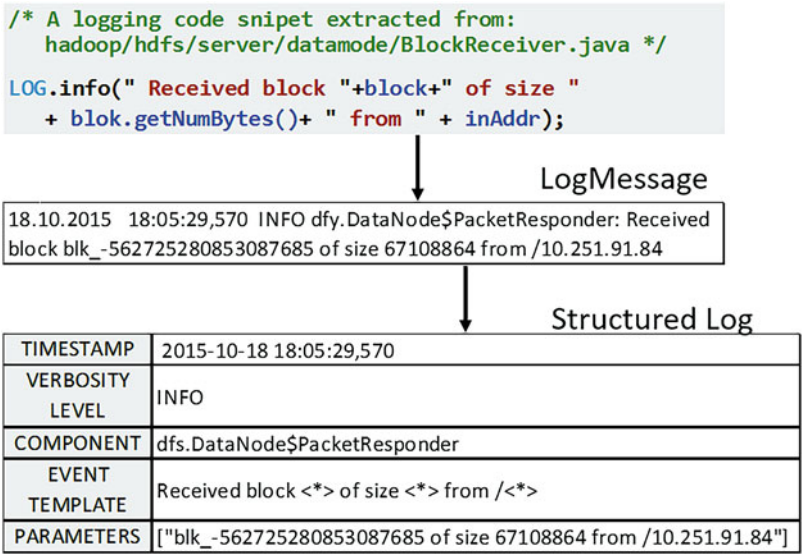


Fig. 1 Example of log parsing according to [40]

have to update the regular expressions after each update of the software, which is complex and error-prone [15].

Both industry and science have developed various concepts for automatic log parsing. In [34] a clustering algorithm based on a frequent pattern matching is presented. Based on this, the *LogCluster* extension is presented in [35]. Another log parser is *Spell*, which calculates the longest common subsequence around the logs to parse as presented in [10]. However, according to [40], *Drain* achieves the best accuracy. We use an implementation of *Drain* by IBM,[1] which is provided as an open-source project. However, when using log parsing, one has to be aware that anomaly detection is affected by log parsing errors, i.e., out-of-vocabulary words and semantic misunderstanding [24]. Our results in Sect. 4.1 are consistent with this.

2.2 Feature Extraction

By parsing the log data, each log entry is assigned a log key of the form E_0, E_1, E_2, E_3, ... The aim of feature extraction is to extract the most important features of the data in order to use it to train a machine learning model; see [14]. The sequence of log templates is then converted to numeric values in order to be processed by

[1] https://github.com/IBM/Drain3.

machine learning models, i.e., in [6]. To do this, each log entry is first replaced by the log key assigned during log parsing. Thus, out of a sequence different log messages a sequence of log keys, i.e., E_1, E_1, E_3, E_2, E_2, E_4, E_2 is created. The log keys are then grouped by using a windowing method. With a windowing method, a log sequence is divided into finitely large partitions as in [29]. While in [29] different types of windowing methods are presented for anomaly detection in logs, only the sliding window method is used here. This way every log message is examined in a different context. The sliding window method is based on the time stamp t of the log entries and a stride h. The time stamp of a log entry indicates when the log entry was created. The increment specifies the time steps to move forward along the time axis; see [6]. In general h is smaller than the fixed window size, which can be a minute or an hour or any other time length or also a certain number of logs. Based on the time window, the chronological sorted log data is grouped. The resulting partitions can now be compared to each other. However, after the log sequences are partitioned in chronological order, they still need to be converted to numeric vectors.

For this, a count matrix X is created. The frequency of each log key is counted in each partitioned log sequence in order to create a count vector; see [14]. For example, if a window with log entries has the form $[E_1, E_0, E_1, E_1, E_2]$, this means that the log key E_0 occurs only once, log key E_1 occurs three times, and log key E_2 occurs again only once. The resulting count vector for the window is then of the form [1, 3, 1], assuming there are only three different log keys. For the construction of the count matrix X, for each log sequence window, a count vector X_i is created such that the entry $X_{i,j}$ stands for the count of the j-th log key in the i-th log sequence window; see [14].

In our work, the vectors X_i will be of dimension 636 for the semi-structured log data in Sect. 3.1, of dimension 212 for the Java exception fingerprints in Sect. 3.2, and of dimension 324 for the structured log data in Sect. 3.3.

2.3 Principal Component Analysis

Principal component analysis (PCA) is a statistical method used to achieve dimension reduction. The underlying idea behind PCA is the projection of high-dimensional data of dimension n into a new space with k dimensions where $k < n$. The newly created k dimensions are called the principal components. The principal components are calculated in such a way that those components are found which have the highest variance in the original high-dimensional data set; see, for example, [14]. As a result, only little information goes lost in the transformation from the n-dimensional space to the k-dimensional space. PCA for anomaly detection in logs is introduced for the first time in [37]. Since then, it has been refined in several ways; see also [20]. In PCA, the data, consisting of a log key count vector, is transformed from an n-dimensional space to a space with k dimensions, where $k < n$. This creates the sub-vector space S_n, which is called the normal subspace. The remaining $n - k$ dimensions form the vector subspace S_a which is denoted as

anomaly subspace; see [37]. Subsequently, the projection of a given log key count vector y into the subspace S_a is calculated by

$$y_a = (1 - PP^T)y, \tag{1}$$

where the matrix $P = [v_1, v_2, \ldots, v_k]$ consists of the first k principal components $v_1, \ldots, v_k$. If the absolute value of the projection $\|y_a\|$ is bigger than a specified threshold value, then y counts as an anomaly; see [14] or [39].

2.4 Clustering

In clustering similar data points are grouped into clusters (see [5]), and clustering algorithms can also be used for anomaly detection. The easiest way is to classify all data points that do not belong to any cluster as an anomaly. Another way to find anomalies with clustering algorithms is to determine the center of the cluster and set a threshold. Every data point that is farther from the center than the specified threshold is considered as an anomaly. The center of a cluster can be determined in different ways; one way is to calculate the average of all data points in the cluster as it is described in [5]. We refer to [22] for different kinds of clustering algorithms in order to find anomalies in log data. Here, *LogCluster* is presented as described in [27]. It uses an agglomerative hierarchical clustering method to detect anomalies in log sequences. The training phase consists of two phases, for which the training data set is divided into two parts. The first phase is called knowledge base initialization. In this phase, the log data is vectorized by creating count vectors as it is described in Sect. 2.2, the clusters are then calculated via a certain clustering algorithm, and the centers of the clusters are also determined; see [14]. Afterward the cosine similarity is determined for each vector to all the other vectors, and with an agglomerative hierarchical clustering method, clusters are formed, as described in [19]. Calculating the similarity of the n-dimensional count vectors X_i and X_j via cosine similarity is done as in the following [27]:

$$Similarity(X_i, X_j) = \frac{X_i \cdot X_j}{\|X_i\| \cdot \|X_j\|} \tag{2}$$

The centers of the clusters are then determined. For each log sequence and thus each count vector X_i, a score calculated as in [27]:

$$Score(i) = \frac{1}{m-1} \sum_{j=1}^{m} \big(1 - Similarity(X_i, X_j)\big) \tag{3}$$

where m stands for the number of all count vectors in a cluster. As a representative for the cluster center, the count vector that achieves the lowest score is selected.

The second phase, known as the online learning phase, is used to fine-tune the created clusters. In this phase, each vector in turn is added to the knowledge. For a given event count vector, the distance to all cluster centers is determined. If the smallest calculated distance is smaller than a fixed threshold, the vector is added to the cluster. If this is not the case, then a new cluster is created; see [14]. The anomaly detection with *LogCluster* takes place by calculating the distance of a given vector with the cluster centers. If the smallest distance is greater than a specified threshold value, the corresponding log sequence is declared as an anomaly.

2.5 Deep Learning

Recurrent neural networks (RNN) are a subset of neural networks which are used to process sequential data; see, for example, [13]. RNN consist of several RNN cells, so-called memory units, that are connected to each other. The output of an RNN memory unit serves as an input for the subsequent cell, making it possible to take past values or past log sequences into account; see [11]. This is essential when processing any kind of sequential data as the output often depends on the previous input, such as in natural language texts, where the meaning of a word also depends on the context. However, simple RNN are rarely used in practice because by the back-propagation in time, the gradient becomes vanishingly small and the model cannot be trained anymore; see [13]. Long short-term memory (LSTM) and gated recurrent unit (GRU) networks solve the vanishing gradient problem and can thus also process longer sequences. RNN, LSTM, and GRU are mainly used when dealing with time series data and also in natural language processing (NLP). In [11], the anomaly detection in log data is therefore presented as an NLP problem, because log data are sequences that are generated with repeating patterns from a computer system. They are subject to grammatical rules that are determined by the program flow of the software; see [11].

In [11], the LSTM model *DeepLog* for anomaly detection in logs is invented; see also [6]. *DeepLog* uses an LSTM with several memory units to generate events based on template sequences to learn the normal execution paths of a computer system. If a log sequence deviates heavily from the pattern the model has learned, the log sequence becomes classified as an anomaly; see [11]. In [11] this model is called *log key anomaly detection model*. In addition to considering only the structured log templates, also the values of the variable part of the event templates might be taken into account, since they also allow conclusions to be drawn about anomalies, for example, the duration of an HTTP call. For this purpose, an LSTM model is trained for each log key; see [11]. In [11], this model is called *parameter value anomaly detection model*. In the following, we explain in more detail how both models are trained.

The *log key anomaly detection model* is used to find anomalies visible in the execution path of a software. The set of distinct log keys, $K = \{E_0, E_1, E_2, \ldots, E_n\}$, is constant for a computer system, which is not undergoing any change. Let m_t be

the value of a log key at position t in a log key sequence. Here, m_t takes one of the n values of K, where the value might depend heavily on its predecessors within the log sequence. The anomaly detection is modeled as a multiclass classification problem, where each log key represents a separate class. A window w of h log keys is used as input for the model s.t.

$$w = \{m_{t-h}, \ldots, m_{t-2}, m_{t-1}\}.$$

The output of the model, a conditional probability for the next log key m_t, is given by

$$Pr\left[m_t = E_i | w\right], \text{ for every } E_i \in K,$$

where $i = 1, \ldots n$. In practice it is possible that several log key values for m_t reflect a normal system behavior. Therefore the log keys in K are sorted based on their probability of occurrence $Pr\left[m_t = E_i | w\right]$. A log key m_t is qualified as normal when it ranges among the top q candidates; otherwise, it is said to be an anomaly; see [11].

As stated in [11], not all anomalies are recognizable by a deviation of the execution path. Therefore, the *parameter value anomaly detection model* detects anomalies contained in the variable parameters of a log message. An example of this could be log messages for HTTP calls, which usually show the call duration in the variable part of the log message. If it differs significantly, in that a call takes significantly longer than usual, it might, for example, indicate that the system is overloaded. In [11] an LSTM model is trained for each log key E_i. A parameter value vector is used as input for the model, which contains all variable values of a log message belonging to log key E_i. The output of the model is a real valued parameter (respectively a vector) which predicts the next value (respectively a vector of the next future values) of the parameter. In the model's training phase, the deviation of the model's prediction with the actual parameter value (respectively vector) is calculated with the mean square error (MSE). After that, a Gaussian distribution is estimated with the calculated MSE values of the validation data set. If a calculated MSE value is outside of a specified confidence interval of the Gaussian distribution, the corresponding parameter (respectively vector) is considered an anomaly, see [11].

3 Log Data Sets

Log data are essential for operators of computer systems as well as for software developers. They are typically semi-structured text strings that can vary a lot from system to system, i.e., [27] or [31]. The variation of the different log formats can even occur within one system, since many computer networks systems use software libraries from third parties, such that open-source projects may be included. This

makes automated analysis from log data even more difficult; see [40]. To describe the different effects of logged states or actions, logging frameworks often define verbosity levels, also called log levels, for log messages as described in [2] or [28]. For instance, when unexpected failures or potential problems occur in a system, logs should be assigned the less verbose levels (FATAL, ERROR, WARN, etc.). On the other side, for the purpose of tracking general information or debugging, the more verbose levels (INFO, DEBUG, TRACE, etc.) are needed. This log level design can assist operators in locating various runtime problems quickly and precisely when auditing highly diverse logs.

A semi-structured log entry consists of two parts [40], a header and a log message. The header contains meta information about a log entry, such as the verbosity level (e.g., INFO, ERROR), a time stamp, or the component that generated the log entry. The log message on the other hand contains also a free text, which can be freely chosen by a developer. The constant part of the log message contains the event template and remains for each event the same. The variable part, on the other hand, contains dynamic information that is generated at runtime, for example, port numbers or IP addresses as in [15]; see also Fig. 1. Many of the anomaly detection models used on log data are evaluated with the publicly available Hadoop Distributed File System (HDFS) data set or the BlueGene/L (BGL) data set, i.e., [14, 23, 25, 36]. These models typically claim very high detection accuracy. For example, most models report an F-measure greater than 0.9 on the HDFS data set. In this work, we want to show that models for anomaly detection perform different, depending on the log data sets they are used for. Or putting it the other way round, preprocessing log data adequately yields better results for some of the anomaly detection models. Instead of using one of the commonly used data sets, the log data examined in this work originate from computer network systems that are developed and operated by a software company itself. The systems are applications which are developed with an internal Java framework. A suitable system installation has been selected such that it:

- has a high volume of log data,
- is a production system that is actively being worked on
- is a multi-client system.

On the other hand, the amount of log data generated per month for the system is limited to approx. 70 million log entries, which makes it possible to handle all the log entries in an almost streaming manner. Furthermore, it is a productive system on which 37 different clients work. We will continue with this installation and refer to it as *Example1*. Furthermore, the utilization due to automated calls from customer systems is very regularly distributed, as can be seen in Fig. 2.

The selected software is called automatically via web interfaces from pre-systems (e.g., SAP systems). For special cases and manual post-processing, a graphical user interface is also available to customers. All log entries are in JSON format. The logs generated are divided into semi-structured logs and structured logs. The semi-structured logs are log entries as described in Fig. 1. They mainly contain information in an unstructured log message. With the structured log entries, on the

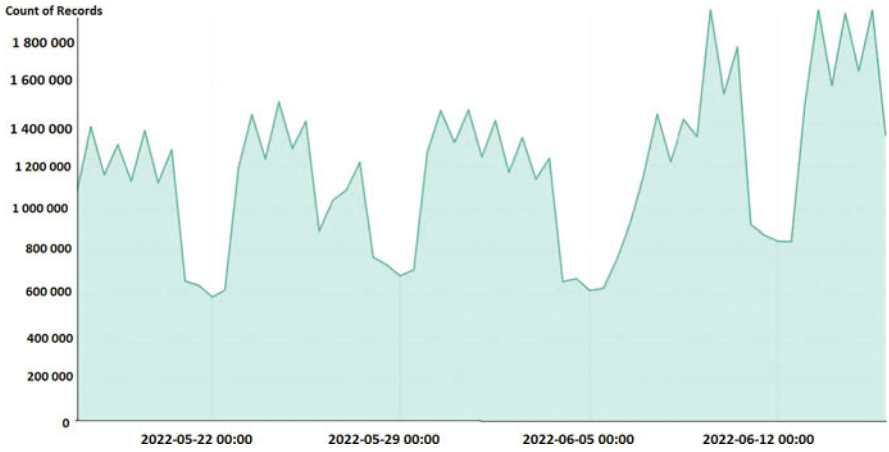


Fig. 2 Number log records per 12h during 30 days of *Example1*

other hand no log message is saved. The authentication log information is stored in key-value pairs as shown in Fig. 8. The exact structure of the two log types as well as the third log type made out of Java fingerprints is explained and described in the next three sections.

3.1 Semi-structured Log Data

An example with the most important fields of a semi-structured log entry of *Example1* is shown in Fig. 3.

The first part of the log entry contains meta information that the logging framework adds automatically, i.e., the verbosity level or the time stamp. In the lower part is the log message and other important fields. The category field describes the type of the log entry, whereby the semi-structured log data is always of the type *event*, contrary to the structured log data which are of one of the types *login*, *gui*, or *access*, and is investigated in Sect. 3.3. For the creation of the training data set, log data from the period of one working week is selected. This results in a data record with approx. 70,000 log entries. The distribution of log verbosity levels is shown in Fig. 4.

Figure 4 shows that the number of logs with log level INFO is most frequently represented in the semi-structured log types. In order to generate event templates out of the semi-structured log data, the log parser *Drain* is used. An open-source implementation of *Drain* is available,[2] and it achieves best results when it is benchmarked with other log parser in [40]. *Drain* generates a fixed-depth parse tree

[2] https://pypi.org/project/drain3/.

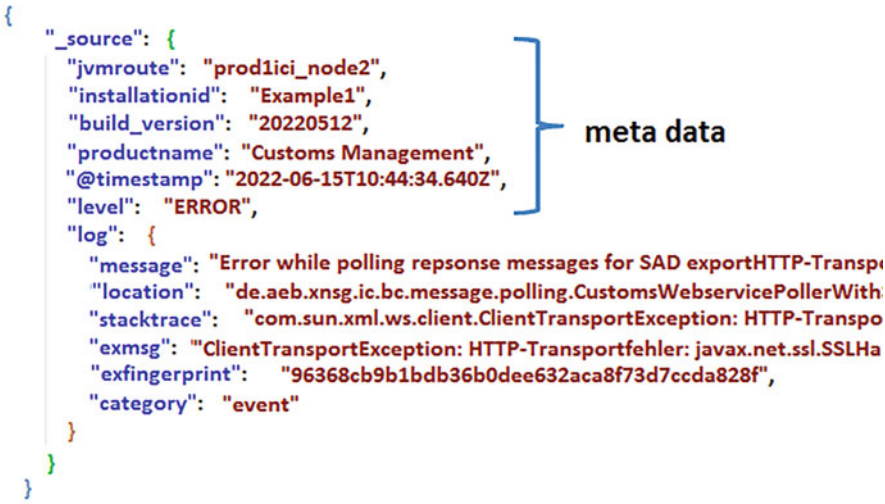


Fig. 3 Structure of a semi-structured log entry used in *Example1*

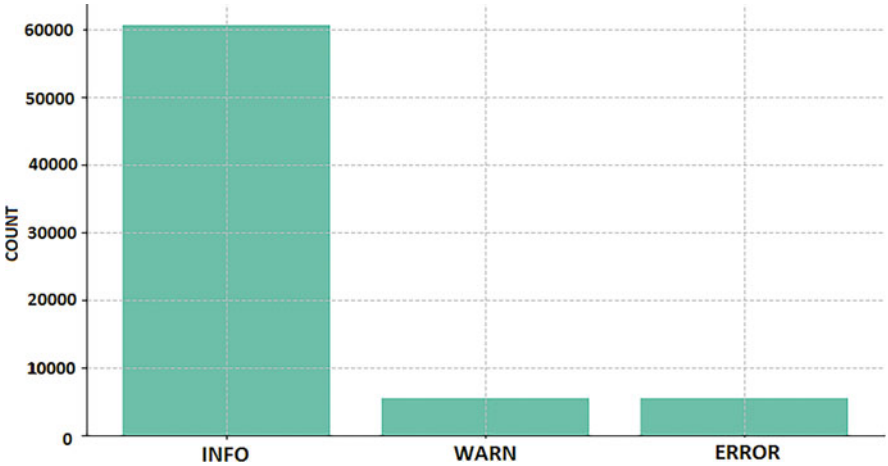


Fig. 4 Distribution of log verbosity levels for installation *Example1*

in order to speed up log parsing. In this work we choose the depth of the tree to be 4 and the maximum number of children of each node in the tree to be 100. *Drain* chooses the appropriate log group for each log event by going through the parse tree and comparing the resulting log messages to an event template belonging to a child group. The similarity of two messages is compared, as long as it is smaller than a certain threshold, which we choose to be 0.4, two messages are grouped in one template. These three hyperparameter values for the parser *Drain*:

- 3 for the depth of the parse tree,

	time stamp	log event	log template	event template ID
6839	2022-05-16 00:00:01.401	executed javascript: /* * Restarts batch sch...	executed javascript: /* * Restarts batch sched...	E_1
6841	2022-05-16 00:00:02.443	executed javascript: /* This script takes all...	executed javascript: /* This script takes all ...	E_6
6133	2022-05-16 00:00:04.902	0 ?de.aeb.xnsg.ic.gb.model.config.ccs.CCSAuth...	0 <:*:> deleted.	E_7
6134	2022-05-16 00:00:04.909	0 Document rule invocation deleted.	0 <:*:> <:*:> <:*:> deleted.	E_8
6117	2022-05-16 00:00:04.933	0 Subscription file deleted.	0 <:*:> <:*:> deleted.	E_9
6135	2022-05-16 00:00:04.943	0 Telemetry report deleted.	0 <:*:> <:*:> deleted.	E_9
6136	2022-05-16 00:00:04.995	0 Short URL deleted.	0 <:*:> <:*:> deleted.	E_9
6118	2022-05-16 00:00:05.268	0 Model deleted.	0 <:*:> deleted.	E_7
6137	2022-05-16 00:00:05.317	0 BI-Journal entry deleted.	0 <:*:> <:*:> deleted.	E_9
6582	2022-05-16 00:00:06.378	0 Goods movement record deleted.	0 <:*:> <:*:> <:*:> deleted.	E_8

Fig. 5 Structure of the semi-structured log data after parsing

```
Caused by: javax.net.ssl.SSLHandshakeException: Remote host terminated the handshake
    at java.base/sun.security.ssl.SSLSocketImpl.handleEOF(SSLSocketImpl.java:1322)
    at java.base/sun.security.ssl.SSLSocketImpl.decode(SSLSocketImpl.java:1160)
    at java.base/sun.security.ssl.SSLSocketImpl.readHandshakeRecord(SSLSocketImpl.java:1063)
    at java.base/sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:402)
    at java.base/sun.net.www.protocol.https.HttpsClient.afterConnect(HttpsClient.java:567)
    at java.base/sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect(AbstractDelegateHttpsURLConnection.java:185)
    at java.base/sun.net.www.protocol.http.HttpURLConnection.getOutputStream0(HttpURLConnection.java:1362)
    at java.base/sun.net.www.protocol.http.HttpURLConnection.getOutputStream(HttpURLConnection.java:1337)
    at java.base/sun.net.www.protocol.https.HttpsURLConnectionImpl.getOutputStream(HttpsURLConnectionImpl.java:246)
    at com.sun.xml.ws.transport.http.client.HttpClientTransport.getOutput(HttpClientTransport.java:89)
```

Fig. 6 Example of a Java stack trace

- 100 for the maximum number of children,
- 0.4 for the similarity threshold,

are determined experimentally, so that only a small number of event templates arises and such that no different event templates are created for the same log messages. After all log messages are parsed, the results are stored in a pandas data frame; see, for example, Fig. 5. The number of event templates or log keys found this way with the log parser *Drain* in the semi-structured data set is 636.

3.2 Java Exception Fingerprints

Java programs generate an exception in the event of an error. This exception is raised in the program stack until it is dealt with. In doing so, a stack trace is created that calls the logs of all methods up to the point at which the exception was generated. The stack trace usually also contains the line numbers of method calls [9]. Next to the stack trace also, an exception message is written, which describes the reason for the error. An example for an exception in Java is shown in Fig. 6.

Here, no unique error numbers are assigned to the Java exceptions. Instead a hash value is assigned to each stack trace, the so-called exception fingerprint. In order to assign the hash value, the variable parts of the exception are discarded. These include the line numbers of the method calls, as well as the exception message, since both of them change when the classes change, which is quite often the case. Are the

```
6138   c659d8c94e8e417092fce69dbeb6ca11818bc1a9
6147   4c16f80c063ea0c96c011b5a20bc26845515893e
6152   da39a3ee5e6b4b0d3255bfef95601890afd80709
6198   c659d8c94e8e417092fce69dbeb6ca11818bc1a9
6233   0a80b93977232f6f2074fd9bf4c942d2445a26cc
6239   c659d8c94e8e417092fce69dbeb6ca11818bc1a9
6328   c84f9c29d7e0717bb148df770aa1e0db6b07ca6d
6398   d9519f9ae5ca8657768bb30a2b51d3b03c7dcd6e
6554   c84f9c29d7e0717bb148df770aa1e0db6b07ca6d
6576   c659d8c94e8e417092fce69dbeb6ca11818bc1a9
```

Fig. 7 Example of a series of Java exception fingerprints

variable parts of the stack trace removed, the exception fingerprint is calculated as a hash value for the remaining part of the stack trace. This exception fingerprint can be used in the same way as an error number. It is similar to a log key, which clearly identifies a log. For an example of a series of exception fingerprints, see Fig. 7. Since our goal is to find errors in the log data that affect the functionality of the system, only log entries with the verbosity level ERROR are selected for the Java exception fingerprints. Since the number of log entries with the verbosity level ERROR is significantly smaller in our data set, we take exception fingerprints from a period of 30 days, such that the size of the resulting data set amounts to about 24,000 log entries.

For the Java exception fingerprints, no further parsing is necessary, since the exception fingerprint serves as an event template ID. This reduces the inaccuracies that arise when parsing the log messages wrongly. The exception fingerprint data record used here contains 212 different exception fingerprints, which corresponds to a number of 212 event template IDs.

3.3 Structured Log Data from Authentication Log Entries

The structured log data are log entries consisting of key-value pairs. They are divided into different categories (i.e., login, access, GUI, etc.) and are designed for clearly defined use cases. An example are the authentication logs for the registration of a user, which log every authentication attempt of a user with duration and type of authentication. All important information for an authentication is saved in explicit fields that can be found for our example in Fig. 8. Structured logs can have both numeric and categorical features. We investigate authentication logs here, since on the one hand, they are by far the most data and, on the other hand, authentication logs could be an indicator that there may be carried out a brute force attack.

Fig. 8 Example of a structured authentication log

Since no log message is logged for these structured log data, the log data is not parsed with a classic log parser. For parsing the structured authentication logs, the categorical features are used, which provide a limited number of different values. This results in a limited number of combinations of the different values for each categorical feature. The possible combinations of the individual values are formed with the cross product and represent the event templates. To keep the number of event templates small, only the relevant features `result`, `auth`, `rolecache`, `type` are used; see Fig. 8. Each of them could indicate a possible attack. If, for example, the categorical feature `result` has the value `ERROR` too often, it could be an indicator of a brute force attack. We determine all distinct feature values for these features over the last 30 days and store them in a data frame. All possible combinations are calculated and result in a data frame with 324 different event templates, which we refer to as the parsing table; see Fig. 9.

4 Anomaly Detection

While using the anomaly detection models of Sect. 2 on the log data described in Sect. 3 several anomalies could be detected, not all of them can be directly linked to an attack of the computer system; there are several other reasons for a log message to become anomalous. We therefore sorted the detected anomalies according to an index in order to make all of the models and data sets comparable. The following

	type	auth	rolecache	result	event template ID
0	BF	DBUSER	True	OK	E0
1	BF	DBUSER	True	ERROR	E1
9	BF	DBUSER	False	OK	E2
10	BF	DBUSER	False	ERROR	E3
81	BF	UNKNOWN	True	OK	E4
82	BF	UNKNOWN	True	ERROR	E5
90	BF	UNKNOWN	False	OK	E6
91	BF	UNKNOWN	False	ERROR	E7
162	BF	ANONYMOUS	True	OK	E8
163	BF	ANONYMOUS	True	ERROR	E9

Fig. 9 Extract of parsing table for the structured authentication log data

table lists and explains the anomalies (Table 1). A detailed description of applying each model on each of the data sets is following in Sects. 4.1, 4.2, and 4.3.

For the implementation of the models of Sect. 2, we use the Python library *loglizer*.[3] For parsing the semi-structured log data, we use the log parser *drain*.[4] All log data is provided in JSON files.

4.1 Semi-structured Log Data

In order to detect anomalies in the semi-structured log data set, the structured log data are grouped in sliding windows of 5 minutes time intervals with a stride of one log entry. This way, one gets as many log sequences as possible for the training process. The resulting log sequences are converted into count vectors, as described in Sect. 2.2, which are used as input for the PCA and the *LogCluster* models. This way 71.935 count vectors are created out of the semi-structured log data set.

A sliding window is also used to create the log sequences for *DeepLog*. However, a constant number of log entries is used here for the window size h, instead of a fixed time length, because the input length h of the window must always remain the same

[3] https://github.com/logpai/loglizer.

[4] https://github.com/IBM/Drain3.

Table 1 List and explanation of detected anomalies

Index	Short description of anomaly	Long description of anomaly
1	Duration of SQL statements too long	This anomaly involves SQL statements that have a long execution time; see, for example, Fig. 10. Some statements take almost 3 seconds to complete. This might indicate an overload to the database, which can lead to system failure
2	Health check of application wrong	Health checks of applications are monitoring certain attributes that are essential for the operation of the application such as messages that could not be processed. An example health check is shown in Fig. 11, where the health check first fails and then the connection is recovered immediately. Since health checks are very often unstable, a machine learning model is needed to recognize only such cases where the health check failure continues for a certain time
3	Authentication too long	If an authentication process lasts more than 10 seconds, a system overload may be the case. This might lead to less customer satisfaction, but may also be an indication of a brute force attack
4	Error writing statistical data	This anomaly states that values for certain metrics, used to monitor the computer system, cannot be saved. This leads to poorer monitoring, which may result in errors that cannot be recognized in time. Usually, the reason for this error is not enough storage space on the hard disk available
5	Software misconfiguration, no responsible person	This anomaly represents a software configuration error. During the configuration of a process, no responsible person was established. While the operation of the software is not endangered, still, it is considered an anomaly since an early detection of the error results in accelerated actions
6	Software misconfiguration, wrong user for authentication	This anomaly is also a misconfiguration of the software. Here, a user does not have sufficient rights for access to the system as, for example, in Fig. 12
7	Connection problems, e-mail server not reached	It is not possible for the system to communicate with the configured e-mail server. As a result, e-mails cannot be sent, and users are not informed about the current processes in the system. The reason could be a network problem or an email server crash
8	Connection problems, external system not available	Here an external system cannot be reached. The cause may also be a network problem
9	Failed download from FTP server, due to wrong authentication data	Due to incorrect credentials used for an FTP server authentication, it is not possible to access the files of the FTP server
10	Connection problems, communication with the message queue disturbed	This anomaly represents a connection problem to a message queue. Due to network problems, messages can no longer be passed to the message queue and thus also not be processed further

45030	INFO	SQL needed 2519 ms for execution (rollback)
45103	INFO	SQL needed 2879 ms for execution (commit)
45031	INFO	SQL needed 2805 ms for execution (commit)
45032	INFO	SQL needed 2586 ms for execution (commit)
45104	INFO	SQL needed 2827 ms for execution (commit)
45105	INFO	SQL needed 2259 ms for execution (rollback)
45106	INFO	SQL needed 2577 ms for execution (rollback)
45070	INFO	SQL needed 2538 ms for execution (rollback)
45071	INFO	SQL needed 2891 ms for execution (commit)
45072	INFO	SQL needed 2907 ms for execution (commit)
45033	INFO	SQL needed 2720 ms for execution (commit)

Fig. 10 Example log sequence of anomaly index 1

34673	WARN	[HEALTHBROKEN] Customs Management->Customs messages->Processing status 'ERROR' ### 6
34674	INFO	[HEALTHHEALED] Customs Management->Customs messages->Processing status 'ERROR' ### 5

Fig. 11 Example log sequence of anomaly index 2

9671	Error while polling response messages for SAD exportHTTP-Transportfehler: java.io.IOException: Unable to tunnel through proxy. Proxy returns "HTTP/1.0 403 Forbidden"
9672	Error while polling response messages for SAD exportHTTP-Transportfehler: java.io.IOException: Unable to tunnel through proxy. Proxy returns "HTTP/1.0 403 Forbidden"
9673	Error while polling response messages for SAD exportHTTP-Transportfehler: java.io.IOException: Unable to tunnel through proxy. Proxy returns "HTTP/1.0 403 Forbidden"

Fig. 12 Example log sequence of anomaly index 6

when dealing with an LSTM model; see Sect. 2.5. The best results according to [11] are received with $h = 10$. The stride is again chosen as one in order to get as many log sequences as possible and in order to be able to measure the dependency of subsequent log sequences. Figure 13 shows the result of the feature extraction. For every log sequence $h = 10$, also the following log entry, m_t is saved. Furthermore, the time stamps of the first and last log entry of the log sequence is saved in order to retrieve it later and to be able to reconstruct it. The conversion of the log sequences into a vector representation is carried out by mapping the log keys to numeric values. Each log key is assigned a numeric value. Then the log key in the log sequence is replaced by the numeric value, resulting in the input vector. This way 71,929 vectors of length 10 are created out of the semi-structured log data set.

4.1.1 Principal Component Analysis

The basic steps of applying PCA to semi-structured log data is illustrated in Fig. 14. In [11] PCA is used as a baseline for all other models. There it is tested on two different data sets, the Hadoop Distributed File System (HDFS) data set and a log

	log key sequence	m	start time	end time
0	[E5, E6, E7, E8, E9, E9, E9, E7, E9, E8]	E10	2022-05-16 00:00:01.401	2022-05-16 00:00:07.795
1	[E6, E7, E8, E9, E9, E9, E7, E9, E8, E10]	E10	2022-05-16 00:00:02.443	2022-05-16 00:00:07.795
2	[E7, E8, E9, E9, E9, E7, E9, E8, E10, E10]	E8	2022-05-16 00:00:04.902	2022-05-16 00:00:09.431
3	[E8, E9, E9, E9, E7, E9, E8, E10, E10, E8]	E7	2022-05-16 00:00:04.909	2022-05-16 00:00:09.611
4	[E9, E9, E9, E7, E9, E8, E10, E10, E8, E7]	E8	2022-05-16 00:00:04.933	2022-05-16 00:00:09.627
5	[E9, E9, E7, E9, E8, E10, E10, E8, E7, E8]	E11	2022-05-16 00:00:04.943	2022-05-16 00:00:09.634
6	[E9, E7, E9, E8, E10, E10, E8, E7, E8, E11]	E7	2022-05-16 00:00:04.995	2022-05-16 00:00:09.654
7	[E7, E9, E8, E10, E10, E8, E7, E8, E11, E7]	E12	2022-05-16 00:00:05.268	2022-05-16 00:00:19.256
8	[E9, E8, E10, E10, E8, E7, E8, E11, E7, E12]	E13	2022-05-16 00:00:05.317	2022-05-16 00:00:19.277
9	[E8, E10, E10, E8, E7, E8, E11, E7, E12, E13]	E14	2022-05-16 00:00:06.378	2022-05-16 00:00:19.301

Fig. 13 Example of log sequences from the semi-structured data set for the *DeepLog* model

data set of an OpenStack system; both data sets are publicly available.[5] One of the main differences of these two data sets compared to the semi-structured log data set under inspection here is the number of resulting log keys. The number of log keys is in the HDFS as well as in the OpenStack data set far below the number of log keys of the semi-structured data set. For the HDFS data set in [11], only 29 log keys are found, for the OpenStack record 40. With the semi-structured data records investigated here $n = 636$, log keys are identified.

The number of the relevant k principal components is calculated to capture 95% of the variance of the data consisting of the counting vectors (see also [37]) and is thus chosen as $k = 272 < n$. To compute the threshold for $\|y_a\|$ marking an anomaly, one can make use of the Q-statistic, a well-known test statistic; see [18]. The computed threshold Q_α guarantees that the false alarm probability is no more than α. For the choice of the confidence parameter α for anomaly detection, we use $\alpha = 0.001$ as in [37], where it is discovered that the results are not sensitive to this parameter choice. Thus, we detect a counting vector y as being an anomaly if

$$\|(1 - PP^T)y\| = \|y_a\| > Q_\alpha = 47.52, \tag{4}$$

where the $n \times k$ matrix P, as in Sect. 2.3, consists of the first k principal components as column vectors.

Using PCA for anomaly detection with the above parameters leads to 11,300 anomalies out of 71,929 semi-structured log entries. The proportion of anomalies found is approximately 15% of the entire data set, which is unrealistic. For this reason, the individual log sequences are not examined in more detail, since it can be assumed that there is a high number of false positives (FP).

In the meta-study [15], PCA is also used to detect anomalies in log data. There, next to the HDFS data set, for which good results are achieved, the Blue Gene/L

[5] https://github.com/logpai/loghub.

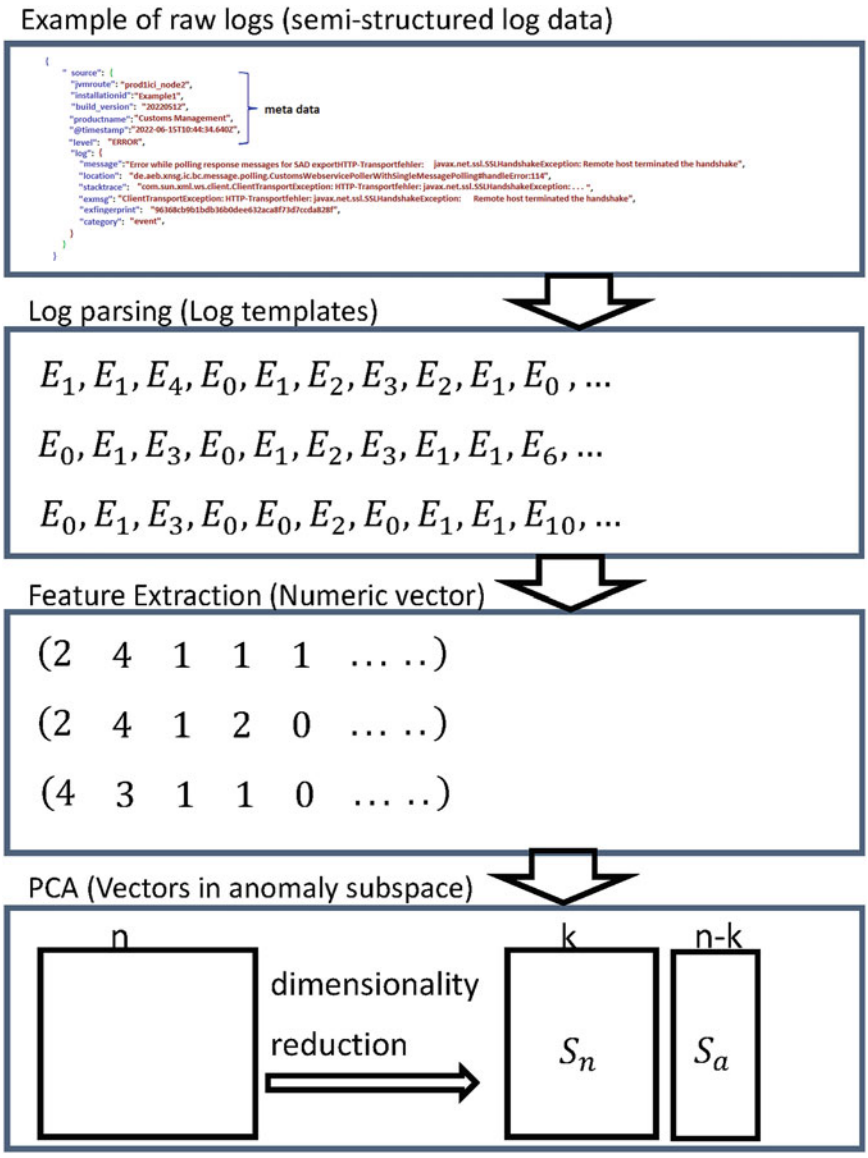


Fig. 14 Steps applying PCA to semi-structured log data

Supercomputer (BGL) data set is investigated for which the results with PCA are also worse. In [15] the authors claim that the data of the BGL data set cannot be separated naturally with a single threshold value. Our results seem to undermine this theory.

4.1.2 Clustering

The *LogCluster* model is used in [15] because it achieves slightly better results than the PCA procedure. The basic steps of applying *LogCluster* to semi-structured log data is illustrated in Fig. 15. For the experimental setup presented here, *LogCluster* models are trained at two different thresholds. The chosen thresholds are 0.3 and 0.5. All evaluated log sequences for the threshold 0.3 can be assigned to one of the first three anomaly indexes from Table 2. For the threshold 0.5 a new index 4 anomaly

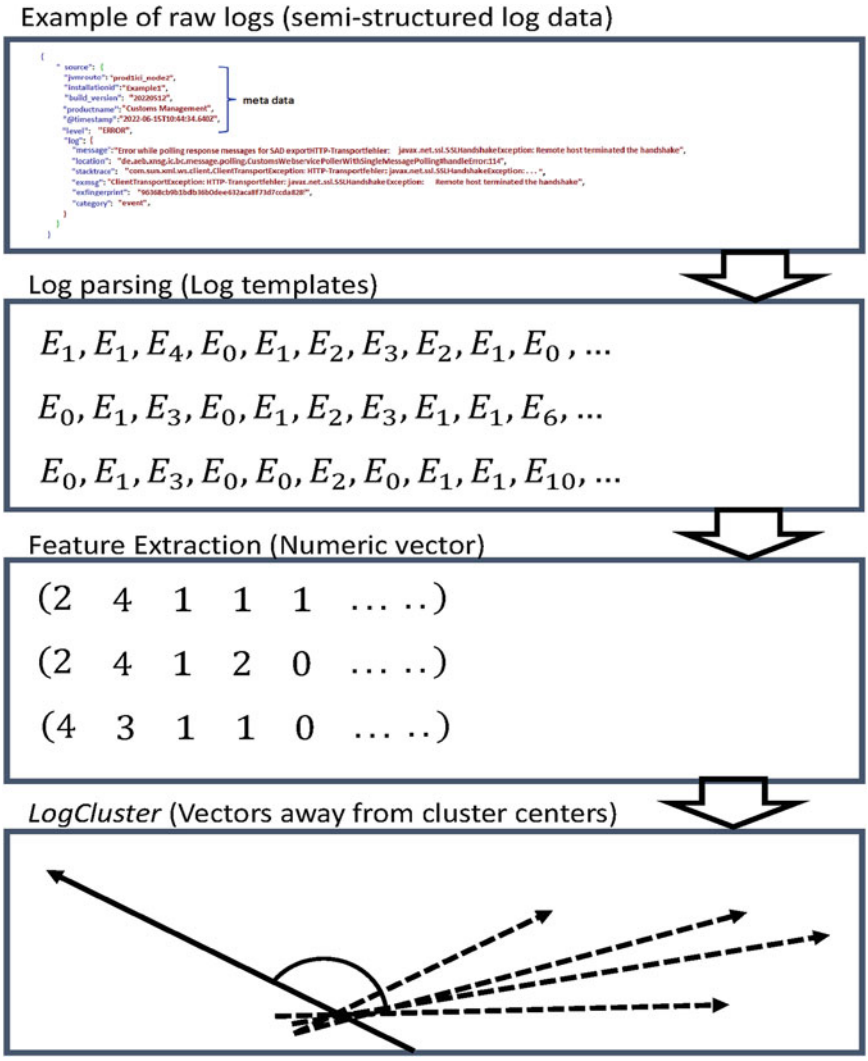


Fig. 15 Steps applying *LogCluster* to semi-structured log data

Table 2 Anomalies with appropriate anomaly index for the semi-structured data set with *LogCluster*

Anomaly index Table 1	Number anomalies	Threshold	Log data index
1	11	0.3	2865, **5322**, 21,539, 33,491, 36,651, **40,402**, **50,548**, **55,119**, 62,101, 65,102, 69,346
2	3	0.3	7268, 9541, 31,073
3	1	0.3	**27,501**
FP	5	0.3	–
1	6	0.5	4527, **40,402**, **50,548**, **55,119**, 62,097, 69,334
2	4	0.5	**5322**, 8840, 33,554, 36,489
3	1	0.5	**27,501**
4	1	0.5	13,709
FP	3	0.5	–

is found, which previously remained undiscovered. The number of representatives anomalous log sequences amounts to 15 for threshold 0.3, respectively, 12 for threshold 0.5. However, 20 respectively 15 anomalies are detected, so that 5 log sequences respectively 3 belong to falls positives (FP). Anomalies appearing under both thresholds are marked bold. Similar anomalies are found in all experiments. The log data index of the found anomalies with the different thresholds are often close together or even overlap.

4.1.3 Deep Learning

As a deep learning method *DeepLog* from [11] is applied to the semi-structured log data. In [11] the model achieves very good results on the HDFS data set. The basic steps of applying *DeepLog* to semi-structured log data is illustrated in Fig. 16. Since our main focus is on detecting anomalous log sequences, only the *log key anomaly detection model* as described in Sect. 2.5 is used in our implementation. The hyperparameters are already optimized in [11] and are listed in Table 3. *DeepLog* works in such a way that it predicts the log key following to an input log sequence. Therefore an accuracy of this prediction might be measured. Accuracy admits the ratio of correctly classified predictions to the total number of all evaluated data points and is calculated by using the true positives (TP), true negatives (TN), false negatives (FN), and the false positives (FP) as follows [17]:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

In our experiments for the semi-structured data set, this accuracy for predicting the next log key amounts to 0.38, which is not very high. Since the prediction of the log

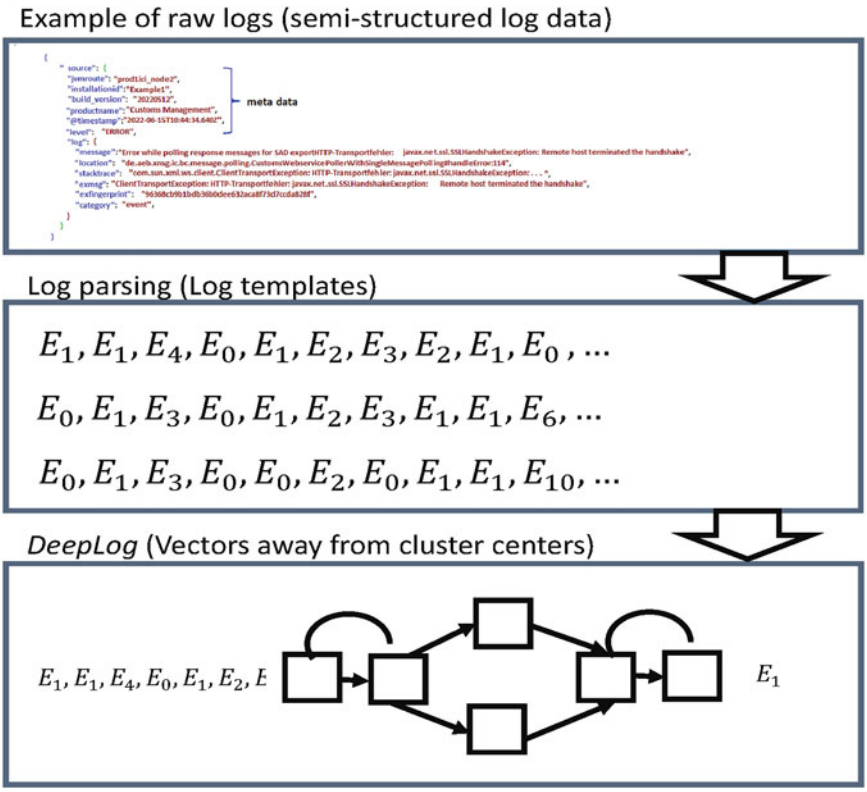


Fig. 16 Steps applying *DeepLog* to semi-structured log data

Table 3 Hyperparameter for the model *DeepLog*

Hyperparameter	Value
LSTM memory units	32
Batch size	32
Training epochs	25
Top candidates q	9

keys is not very good, also the anomaly detection will detect a lot of FP. Similar as with PCA, *DeepLog* detects 16,296 anomalies, which amounts to almost 15% of all the log sequences. Since the number of anomalies found is very high, the individual log sequences, as in Sect. 4.1.1, are not examined in detail. One reason for this might be as in Sect. 4.1.1 that by parsing the log sequences a high number of log keys is generated. In [11] a significantly smaller number of log keys is generated for the data sets.

Since the number of log keys seems to play an important role in detecting anomalies with PCA or with *DeepLog* and since a high number of log keys can be better managed with a clustering method, for the semi-structured data set, *LogCluster* outperforms the other two methods.

4.2 Java Exception Fingerprints

The exception fingerprint data record contains 212 different exception fingerprints, which corresponds to a number of 212 event template IDs. The procedure for vectorizing the hash values belonging to the exception fingerprints is analogous as for the semi-structured data set in Sect. 4.1. Only the selected hyperparameters change. As there are significantly fewer log entries, because only the logs with verbosity level ERROR are used, the window size is selected here at 180 minutes. This results in a data set with 24,013 log entries. For the training data set for the *DeepLog* model, the same parameters are used as in Sect. 4.1 for the semi-structured data set, such that a window of log size $h = 10$ is used. This results in 24,004 log sequences.

4.2.1 Principal Component Analysis

The number of the relevant k principal components is as in Sect. 4.1.1 calculated to capture 95% of the variance of the data consisting of the counting vectors; it is therefore $k = 92$. The threshold $Q_\alpha = 20.21$ is also chosen as in Sect. 4.1.1.

This leads to 2020 anomalies out of 24,007 log entries. Again the proportion of anomalies found is very high and unrealistic. For this reason the individual log sequences are not examined in more detail, since it can be assumed that there is a high number of FP. The reasons for this are probably the same as for the semi-structured data records in Sect. 4.1.1. It might be due to the fact that, as in [15], the data set is not separable with a single threshold.

4.2.2 Clustering

The results here are obtained using the *LogCluster* method, as presented in [15], for the Java exception fingerprint record. Again the thresholds 0.3 and 0.5 are tested. The results can be found in Table 4. The number of anomalies found with a threshold of 0.5 is significantly lower than with the threshold value 0.3. Only six anomalies are detected for the threshold 0.5. However, a new anomaly of index 10 is recognized here; see Table 4. Again anomalies detected with both thresholds are written in bold letters. In the case of the Java exception fingerprint records, only log entries with the verbosity level ERROR are considered. However, not all detected anomalies are real anomalies, but there exist also FP. The log sequences, which are declared as FP, contain technical errors that do not significantly affect the operation of the software, for example, incorrect entries by users. These log entries should not have the verbosity level ERROR, but WARN.

Table 4 Anomalies with appropriate anomaly index for Java exception fingerprints with *LogCluster*

Anomaly index Table 1	Number anomalies	Threshold	Log data index
5	5	0.3	4157, **5932**, 7163, 8690, **9361**
6	2	0.3	9980, 10,160
7	1	0.3	18,772
8	1	0.3	19,863
9	2	0.3	22,398, 23,703
FP	4	0.3	–
5	2	0.5	**5932**, **9361**
7	1	0.5	20,300
10	1	0.5	19,113
FP	2	0.5	–

Table 5 Number of authentication data from client *A*, *B* and *C*

Client	Log sequences
A	261,529
B	413,041
C	263,858

4.2.3 Deep Learning

As in Sect. 4.1.3 the deep learning method *DeepLog*is applied to the Java exception fingerprints with the same hyperparameters. For the Java exception fingerprints, the accuracy amounts to 0.78 which is a lot better as the accuracy in Sect. 4.1.3. However, the number of anomalies detected amounts to 882, which is relative to the total number of log sequences too high. For this reason, the individual log sequences are not examined in more detail, since it can be assumed that there are a large number of FP.

4.3 Structured Log Data from Authentication Log Entries

The frequency of authentication logs is very high, which is why a smaller window size for corresponding log entries is sufficient. For analyzing PCA and a clustering method, a window size of 5 minutes is therefore chosen as well as a stride of one. The authentication log data of three different client applications are evaluated; see Table 5. When examining the authentication logs, one might be able to detect brute force attacks. The original authentication logs used here, however, do not show any failed authentication attempts. Because of this, synthetic anomalies are created. For this, a training data set without anomalies is created as well a test data set containing simulated attacks. The ratio of training data to test data is 75% training data and

25% test data. Through the artificial inserted failed authentication attempts, the log records can be labeled, thereby calculating of evaluation metrics becomes possible and the models can be evaluated more easily.

For simulating a brute force attack, it is assumed that in an attack multiple authentication attempts in a row fail. A failed authentication attempt can be recognized by the fact that the verbosity level contains the value ERROR. An attack or an anomaly is present if there are more than one failed attempt in a row. For creating synthetic anomalies, 5% of the test data are randomly selected, and the value of the verbosity level is set to ERROR for all log entries in the selected log sequence. The manipulated log sequences receive the label 1, which stands for an anomaly. All other log sequences are given the label 0 and stand for a normal log sequence.

For the supervised evaluation of the anomaly detection, the performance metrics *precision*, *recall*, and *F-score* are used:

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

This way a small value for precision stands for a small number of FP:

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

A high recall therefore means a low number of FN values. The F-score is the harmonic mean between precision and recall and is calculated as follows (see [13]):

$$F\text{-score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{8}$$

The structured log data from authentication log entries is only evaluated with PCA and clustering, since as Sects. 4.3.1 and 4.3.2 are going to show, the results there are already very good, so that the higher effort of a deep learning model is no longer justified.

4.3.1 Principal Component Analysis

In [14] an F-score of 0.79 for the HDFS data set and for the BGL data set of 0.55 is achieved. For the structured data set of authentication log entries, we achieve the following values, shown in Table 6, where the number k of principal components and the threshold level are calculated as in Sect. 4.1.1.

One reason for the significantly better F-scores in the authentication log data set is the homogeneity of the log data. On the one hand, the number 22 of different log keys is very low. Furthermore, since the different systems are called automatically by upstream systems, the frequency with which the individual log entries are

Table 6 Precision, Recall and F-score for the structured authentication logs evaluated with PCA

Client	Threshold	k	Precision	Recall	F-score
A	0.59	4	1	0.98	0.99
B	2.18	4	1	0.97	0.98
C	0.99	2	1	0.96	0.98

Table 7 Precision, Recall and F-score for the structured authentication logs evaluated with *LogCluster*

Client	Threshold	Precision	Recall	F-score
A	0.3	0.99	1	0.99
A	0.5	0.99	1	0.99
B	0.3	1	0.99	0.99
B	0.5	1	1	1
C	0.3	1	0.99	0.99
C	0.5	0.99	1	0.99

created is very uniform. This allows a model to learn the normal authentication log operations very well and also recognize the deviations well.

4.3.2 Clustering

Very good values for the F-score are also achieved with the *LogCluster* method, as it can be seen from Table 7. The F-scores obtained with the *LogCluster* method are clearly better than in [14]. In [14] the *LogCluster* method achieves for the HDFS data set an F-score of 0.8 and for the BGL data set an F-score of 0.57. The reason for the good performance is also the homogeneity of the data.

The experiments carried out with the authentication log data set show that the methods PCA and *LogCluster* can be used very well on homogeneous data. As the authentication log data is one example of structured log data, as presented in Sect. 3.3, the experiments performed here could be transferred to the other types of structured log data.

5 Conclusion

While examining three different log data sets, a semi-structured data set, the Java exception fingerprint records, and the structured authentication log records, different machine learning models for anomaly detection are tested in order to identify whether something is going wrong on a computer system. The three models PCA from [37], *LogCluster* from [27], and *DeepLog* from [11] belong to the models which are benchmarked most often in publications; see [23]. It turns out

that the clustering method *LogCluster* performs best in all cases evaluated here. Furthermore, one sees that the more structured the original training data set is, the better the results for detecting anomalies in the log data. We also show that elaborate and complicated deep learning models in these cases do not obtain the best results and are unnecessary the more structured the log data is. However, a lot of the false positives obtained with DeepLog and PCA might be due to parsing errors and parsing noise. Thus, using anomaly detection models which do not depend on a parser as in [24] might lead to better results. Thus, we also propose that more adequate, publicly available and diverse log data sets are needed to ensure the applicability of deep learning in anomaly detection for log data. Otherwise simple, fast and conventional clustering methods as for example *LogCluster* remain superior.

References

1. Aggarwal, C.C.: Outlier Analysis. Springer, Cham (2017)
2. Anu, H., Chen, J., Shi, W., Hou, J., Liang, B., Qin, B.: An approach to recommendation of verbosity log levels based on logging intention. In: IEEE International Conference on Software Maintenance and Evolution (ICSME), Cleveland, OH, USA, pp. 125–134 (2019). http://doi.org/10.1109/ICSME.2019.00022
3. Bodik, P., Goldszmidt, M., Fox, A., Woodard, D.B., Andersen H.: Fingerprinting the datacenter: automated classification of performance crises. In: Proceedings of the 5th European conference on Computer systems, pp. 111–124 (2010). http://doi.org/10.1145/1755913.1755926
4. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, pp. 25–29 (2006). http://doi.org/10.1145/1143844.1143865
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. (2009). http://doi.org/10.1145/1541880.1541882
6. Chen, Z., Liu, J., Gu, W., Su, Y., Lyu, M.R.: Experience Report: Deep Learning-Based System Log Analysis for Anomaly Detection (2022). http://doi.org/10.48550/arXiv.2107.05908
7. Chen, M., Zheng, A., Lloyd, J., Jordan, M., Brewer, E.: Failure diagnosis using decision trees. In: International Conference on Autonomic Computing, New York, NY, USA, pp. 36–43. IEEE (2004). http://doi.org/10.1109/ICAC.2004.1301345
8. Chen, X., Lu, C., Pattabiraman, K.: Predicting job completion times using system logs in supercomputing clusters. In: DSN-W13, Proc. of the 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop, pp. 1–8. IEEE (2013). http://doi.org/10.1109/ISSRE.2016.21
9. Darwin, I.F. : Java Cookbook. O'reilly Media, Sebastopol (2020)
10. Du, M., Li, F.: Spell: Streaming parsing of system event logs. In: IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, pp. 859–864 (2016). http://doi.org/10.1109/ICDM.2016.0103
11. Du, M., Li, F., Zheng, G., Srikumar, V.: DeepLog: anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1285–1298 (2017). http://doi.org/10.1145/3133956.3134015
12. Guo, H., Yuan, S., Wu, X.: LogBERT: log anomaly detection via BERT. In: 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, pp. 1–8 (2021). http://doi.org/10.1109/IJCNN52387.2021.9534113

13. Goodfellow, I., Bengio, Y., Courville, A.. Deep Learning. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, MA (2016)
14. He, S., Zhu, J., He, P., Lyu, M.R.: Experience report: system log analysis for anomaly detection. In: 2IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), Ottawa, ON, Canada, pp. 207–218 (2016). http://doi.org/10.1109/ISSRE.2016.21
15. He, S., Zhu, J., Zheng, Z., Lyu, M.R.: Drain: an online log parsing approach with fixed depth tree. In: IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, pp. 33–40 (2017). http://doi.org/10.1109/ICWS.2017.13
16. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. **22**, 85–126 (2004). http://doi.org/10.1007/s10462-004-4304-y
17. Hossin, M., Sulaiman, M.N.: A review on evaluation metrics for data classification evaluations. Int. J. Data Mining Knowl. Manag. Process **5**, 1–11 (2015). http://doi.org/10.5121/ijdkp.2015.5201
18. Jackson, J.E., Mudholkar, G.S.: Control procedures for residuals associated with principal component analysis. Technometrics **21**, 341–349 (1979)
19. James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning, 2nd edn. Springer, New York (2021)
20. Juvonen, A., Sipola, T., Hämäläinen, T.: Online anomaly detection using dimensionality reduction techniques for HTTP log analysis. Comput. Networks **91**, 46–56 (2015). http://doi.org/10.1016/j.comnet.2015.07.019
21. Kwon, D., Kim, H., Kim, J., Suh, S.C., Kim, I., Kim, K.J.: A survey of deep learning-based network anomaly detection. Cluster Comput. **22**, 5949–5961 (2019)
22. Landauer, M., Skopik, F., Wurzenberger, M., Rauber, A.: System log clustering approaches for cyber security applications. Comput. Secur. **92** (2022). http://doi.org/10.1016/j.cose.2020.101739
23. Landauer, M., Onder, S., Skopik, F., Wurzenberger, M.: Deep learning for anomaly detection in log data: a survey. Mach. Learn. Appl. **12**, 1–19 (2023). http://doi.org/10.1016/j.mlwa.2023.100470
24. Le, V.H., Zhang, H.: Log-based anomaly detection without log parsing. In: 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, Australia, pp. 492–504 (2021). http://doi.org/10.1109/ASE51524.2021.9678773
25. Le, V.H., Zhang, H.: Log-based anomaly detection with deep learning: how far are we? In: Proceedings of the 44th International Conference on Software Engineering (ICSE '22), Pittsburgh, Pennsylvania, pp. 1356–1367 (2022)
26. Liang, Y., Zhang, Y., Xiong, H., Sahoo, R.: Failure prediction in IBM Bluegene/l event logs. In: 7th IEEE International Conference on Data Mining (ICDM 2007), pp. 583–588. IEEE (2007)
27. Lin, Q., Zhang, H., Lou, J.-G., Zhang, Y, Chen, X.: Log clustering based problem identification for online service systems. In: 38th International Conference on Software Engineering Companion (ICSE-C), Austin, TX, pp. 102–111 (2016)
28. Mendes, E., Petrillo, F.: Log severity levels matter: a multivocal mapping. In: IEEE 21st International Conference on Software Quality, Reliability and Security (QRS), Hainan, China, pp. 1002–1013 (2021). http://doi.org/10.1109/QRS54544.2021.00109
29. Pal, G., Li, G., Atkinson, K.: Big data real time ingestion and machine learning. In: IEEE Second International Conference on Data Stream Mining and Processing (DSMP), Lviv, Ukraine, pp. 25–31 (2018). http://doi.org/10.1109/DSMP.2018.8478598
30. Pang, G., Chunhua, S., Longbing, C., Van Den Hengel, A.: Deep learning for anomaly detection: a review. ACM Comput. Surv. (2021). http://doi.org/10.1145/3439950
31. Ryciak, P., Wasielewska, K., Janicki, A.: Anomaly detection in log files using selected natrual language processing methods. Appl. Sci. **12** (2022). http://doi.org/10.3390/app12105089
32. Šabić, E., Keeley, D., Henderson, B., Nannemann, S.: Healthcare and anomaly detection: using machine learning to predict anomalies in heart rate data. AI & SOCIETY **121** 149–158 (2021). http://doi.org/10.1007/s00146-020-00985-1
33. Schneider, P., Xhafa, F.: Anomaly Detection and Complex Event Processing over IoT Data Streams. Academic Press, Cambridge, MA (2022)

34. Vaarandi, R.: A data clustering algorithm for mining patterns from event logs. In: Proceedings of the 3rd IEEE Workshop on IP Operations and Management (IPOM 2003) (IEEE Cat. No.03EX764), Kansas City, MO, USA, pp. 119–126 (2003). http://doi.org/10.1109/IPOM.2003.1251233
35. Vaarandi, R., Pihelgas, M.: LogCluster - a data clustering and pattern mining algorithm for event logs. In: 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain, pp. 1–7 (2015). http://doi.org/10.1109/CNSM.2015.7367331
36. Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P.K., Alfarraj, O., Tolba, A.: LogEvent2vec: LogEvent-to-Vector based anomaly detection for large-scale logs in Internet of Things. Sensors **9** (2020). http://doi.org/10.3390/s20092451
37. Xu, W., Huang, L., Fox, A., Patterson, D., Jordan, M.I.: Detecting large-scale system problems by mining console logs. In: Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles (SOSP '09). Association for Computing Machinery, New York, NY, USA, pp. 117–132 (2009). http://doi.org/10.1145/1629575.1629587
38. Zhao, Z., Xu, C., Li, B.: A LSTM-based anomaly detection model for log analysis. J. Sign. Process Syst. **93**, 745–751 (2021). http://doi.org/10.1007/s11265-021-01644-4
39. Zhang, B., Yang, J., Wu, J., Qin, D., Gao, L.: PCA-subspace method – Is it good enough for network-wide anomaly detection. In: IEEE Network Operations and Management Symposium, Maui, HI, USA, pp. 359–367 (2012). http://doi.org/10.1109/NOMS.2012.6211919
40. Zhu, J., He, S., Liu, J., He, P., Xie, Q., Zheng, Z., Lyu, M.R.: Tools and Benchmarks for Automated Log Parsing. IEEE Press, Montreal, QC (2019). http://doi.org/10.1109/ICSE-SEIP.2019.00021

Detecting Web Application DAST Attacks in Large-Scale Event Data

Pojan Shahrivar and Stuart Millar

1 Introduction

The advent and expansion of the cloud has seen web applications flourish in every corner of our digital lives. From email and e-commerce to banking and social networks, these platform-agnostic experiences have become essentials of everyday life. However, our increasing reliance on these applications has also made them prime targets for cybercriminals. In fact, we have seen a significant surge in web application attacks in recent years. [4].

The OWASP Top Ten [29] is a useful consensus list of the most critical vulnerabilities in web applications, such as broken access control and injections. In reality though, securing and protecting web application servers is inherently difficult since they have to serve legitimate requests while simultaneously denying access to malicious ones. The distinction between a malicious and legitimate request can be difficult to make. Moreover, accidentally blocking legitimate traffic can have serious consequences, potentially disrupting operations and damaging trust with users. With web attacks being the most common form of compromise [12], there is a pressing need for defensive capabilities to protect businesses, consumers, and governments.

Dynamic application security testing (DAST) gathers information on a web app's potential vulnerabilities by sending a variety of HTTP requests in a brute-force manner through automated pen testing techniques, which can then be remediated in the software development lifecycle. There is a valid global commercial industry for these tools; plus some are freely open source [9, 18, 30]. However, DAST scanners are also used maliciously by skilled and unskilled attackers. The latter, 'script kids', may use scanners before selling their findings to a more proficient actor [7]. Hence,

P. Shahrivar (✉) · S. Millar
Rapid7, LLC, Boston, MA, USA
e-mail: pojan_shahrivar@rapid7.com; stuart_millar@rapid7.com

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_14

failure to detect and block DAST activity could expose an organisation to more severe abuse by sophisticated adversaries. Since the adversary's attack capacity is limited only by the available bandwidth and CPU, defensive systems need to prevent DAST scanning attacks with minimal manual intervention. In a production environment with multiple apps and millions of events, it is not feasible to check each attack alert by hand. With traditional detectors based on statistical thresholds suffering over-optimisation and excess false positives, it is essential to develop new protection mechanisms against malicious DAST activity.

An effective existing approach to securing web apps more generally is to run separate, dedicated security solutions in a layered fashion to filter incoming traffic. One such protection mechanism is a web application firewall (WAF). A WAF operates in the application layer and protects the web application by analysing each HTTP request and then filtering, monitoring, and blocking malicious traffic. WAFs have evolved into next-generation WAFs (NGWAFs) to include sophisticated event analysis engines that detect and prevent other malicious attacks such as credential tampering and distributed denial of service (DDoS). This motivates us to propose an automated machine learning (ML) classifier to augment an NGWAF and block actors performing DAST scanning attacks while avoiding the arguably arbitrary adjustments of threshold-based methods.

In this chapter, we first illustrate a method of preprocessing millions of web application events in a temporal fashion using a tumbling window approach to create a proprietary dataset. Then empirically demonstrated a random forest ML model with an optimal window size $w = 60$ seconds achieves an F1 score of 0.94 and a miss rate of 6% in detecting DAST attacks on average across three production-grade web apps.

2 Related Work

2.1 Traditional Threshold-Based Detection Approaches

In the past, malicious activity was often detected using threshold-based models. These models characterised client behaviour through statistical models, heuristics, and anomaly detection rules [3, 11, 17, 32]. For example, a model designed to classify login attacks might set thresholds for the number of hourly attempts to input credentials and the number of IP addresses that an actor uses. These thresholds could even adapt over time, such as by calculating a moving average of the IPs per user.

Pioneering research from Denning [3] revealed a correlation between anomalous activity and misuse, employing statistical techniques to compare user behaviour against both static and dynamic norms. This study emphasised the need for greater network security and inspired other researchers to delve deeper into the topic.

Kruegel [11] proposed an intrusion detection system (IDS) that detected attacks on web servers and applications by analysing the statistical characteristics of HTTP

traffic and the parameters within client-server queries. Perez [17] used Markov chains to construct a web intrusion detection mechanism. In a similar vein, Torrano [32] presented an anomaly detection model for HTTP traffic coming from and to a web application, defining the standard behaviour of the application through statistical properties.

Although these threshold-based systems might appear simple to set up, choosing the right thresholds can be challenging. The optimal thresholds will likely differ among various web applications. Additionally, certain situations may require multivariate thresholds. For instance, if a username pops up in different countries, but always with the same browser fingerprint, it could suggest that a VPN is being used and the activity is not suspicious. To remain effective, the threshold-based system would then need to consider the number of unique browser fingerprints. As these complexities rise, the heuristics become more difficult to implement, potentially requiring more manual interventions.

The selection of a suitable threshold is further complicated by the noise that these complexities generate. Too low a threshold increases the detection rate but also produces time-consuming false positives. On the other hand, a threshold set too high might overlook some malicious activity. These combined factors make it challenging to establish effective rules and thresholds.

In contrast, our proposed machine learning (ML) classifier avoids thresholds, instead focusing on temporal features that suggest DAST scanning attacks based on tumbling time windows. Notably, while there has been extensive research on the detection of network layer attacks has been conducted, work specifically related to the application layer is less prevalent and less developed [22].

2.2 Machine Learning Detection Approaches

The rise of machine learning techniques inspired researchers to develop unsupervised anomaly-based network intrusion detection models [8, 25, 36, 38]. Javaid et al. [8] applied self-taught learning, a deep learning-based technique, to NSL-KDD [28], a dataset designed to address certain limitations of the KDD'99 dataset [34]. Shone et al. [25] introduced a non-symmetric deep autoencoder (NDAE) for unsupervised feature learning and proposed a model built using stacked NDAEs, evaluated with the benchmark KDD'99 and NSL-KDD datasets. Vinayakumar et al. [36] conducted a study on the effectiveness of various ML methods in detecting future cyberattacks, conducting experiments with both traditional ML algorithms and neural networks using benchmark datasets such as KDD'99 and NSL-KDD.

Researchers such as Pan et al. [14] and Sun et al. [27] have explored the feasibility of unsupervised and semi-supervised approaches for web attack detection. The data they used were based on a monitoring tool that captures the behaviour of the web application by extracting traces of program execution from the running software. The monitoring model was created using supervised training with test suites developed as part of the software development process. Other researchers [2]

proposed using ML to model the normal behaviour of applications and to detect cyberattacks. However, it was unclear to what extent ML was actually used in the solution, as they also used regular expressions and dynamic programming.

In our view, the use of machine learning in enterprise application security is an area ripe for exploration. To the best of our knowledge, no one has directly addressed the detection of DAST vulnerability scanning attacks, probably due to the lack of relevant web application datasets and the need for expensive domain knowledge. We see an opportunity to fill this gap in research by using a large, real-world dataset to specifically investigate the detection of DAST attacks at the application layer using machine learning.

3 Methodology

In this chapter, we will introduce a machine learning (ML) classifier that we have designed to detect malicious activity from dynamic application security testing (DAST) scanners. This classifier will enhance the capabilities of a next-generation web application firewall (NGWAF), forming a key part of a broader security system. Once a source IP has been identified as a potential threat, it can be blocked to halt further scanning attempts.

We categorise IP activity into two types: 'positive', which means it is originating from a DAST scanner, and 'negative', which means it is not associated with such a scanner. It is important to keep the number of false-positive detections to a minimum to avoid blocking legitimate requests.

In the following sections, we will discuss how we curated raw event data, provide an overview of the NGWAF, and delve into the concepts of tumbling windows, data analysis, and feature engineering. To help guide our research, we have made a few assumptions:

- The adversary we are dealing with is not particularly skilled and cannot write custom payloads, leading them to use a DAST scanner for malicious purposes.
- Attacks are not distributed and come from the same source IP.
- The DAST scanner is used in its default configuration, without any custom APIs or extensions.

3.1 Data Curation

The AppSensor framework, an industry standard from OWASP [31], provides a structure for the implementation of intrusion detection and automated response at the application layer. This framework is fundamentally defensive, aiming to detect malicious actors rather than discovering vulnerabilities in a web app.

Table 1 AppSensor event features

Feature	Description
`ts`	Event timestamp
`ip`	Source IP
`response-code`	Response code (e.g. `200`, `404`, etc.)
`payload`	Payload (e.g. `OR 1=1`)
`method`	Method (e.g. `POST`, `GET`, etc.)
`headers`	Headers in the request
`parameter`	Parameter set in the URL
`sensor`	Triggered sensor
`pattern`	Triggered sensor pattern
`location`	Location (body, cookie, header, or query)
`uri`	URI (e.g. https://www.space-travel.com/api/saturn/)

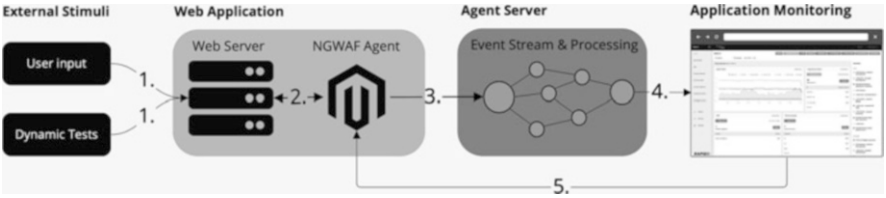


Fig. 1 Diagram of the NGWAF architecture, depicting the integration and process from stimuli to security policy implementation

In our next-generation web application firewall (NGWAF), we utilise the AppSensor framework through a series of detection points or sensors, which are activated by suspicious activities. These could include a range of actions from attackers, such as sending an SQL injection payload, attempting to bypass authentication, malforming requests with unusual encodings, or engaging in credential stuffing, among others. When these activities occur, they trigger a regular expression pattern in the relevant sensor, which in turn creates a time-stamped AppSensor event. Table 1 provides a list of features associated with an AppSensor event. Given the nature of web traffic, there can be millions of such events in a single 24-hour period, providing a rich and timely source of raw data from the real world.

As shown in Fig. 1, the NGWAF comprises a monitoring agent, an event analysis engine, and an application monitoring user interface for a threat overview. The process from receiving external stimuli (Step 1) to recording an event, processing it, and ultimately implementing a new security policy is described in Steps 2–5. The event analysis engine consumes the events generated by the agent and uses policy scripts to automate actions in response to the observed signals. The agent, integrated into an application or its runtime environment, monitors, detects, and prevents attacks in real time.

Table 2 Sensors implemented in the NGWAF

Sensor	Description
`cmdi`	Command injection
`excsrf`	Cross-site and anti-request forgery tokens
`exsql`	SQL exceptions
`fpt`	File path traversal
`null`	Embedding null code
`reqsz`	Unusual request size
`retr`	Line-break character
`rspsz`	Unusual response size
`s4xx`	All response codes 4xx
`s5xx`	All response codes 5xx
`sqli`	SQL injection
`uaempty`	User agent empty
`xss`	Cross-site scripting
`xxe`	XML external entity processing

Policies are crafted by administrators based on insights gleaned from the application monitoring interface. Depending on the specifics of the policy, the engine discerns how to manage communications, which can include permitting, documenting, or blocking an IP address. Presently, administrators have the power to block IP addresses that have launched DAST attacks by perusing the user interface. However, this process is potentially laborious and costly due to the sheer volume of incidents, leading to cognitive stress and the risk of alert fatigue.

Following the principles of the AppSensor framework, each attack category has multiple detection points within the NGWAF, which function as sensors (see Table 2 for details). The NGWAF employs deep packet inspection (DPI) to implement the AppSensor framework, capturing details of activity that originates within the application layer. DPI bolsters the ability of web applications to filter suspicious request and response activity by scrutinising strings within HTTP requests and comparing these patterns against a broad spectrum of potential attacks.

When a sensor is activated, an AppSensor event is produced and sent to the event stream, also known as the AppSensor stream. We gathered and stored the streams from our three most trafficked production web applications over a 48-hour period, a process we will discuss more thoroughly in Sect. 4.1.

3.2 Tumbling Windows

DAST scanners typically operate in a conspicuously overt brute-force manner. Some of the most discernible patterns include the rate of requests and the diversity of requests originating from a source IP. To capitalise on this, we perform an aggregation of events over time, thereby creating ‘tumbling windows’. By initially grouping events from the same source IP address, we can construct numerical

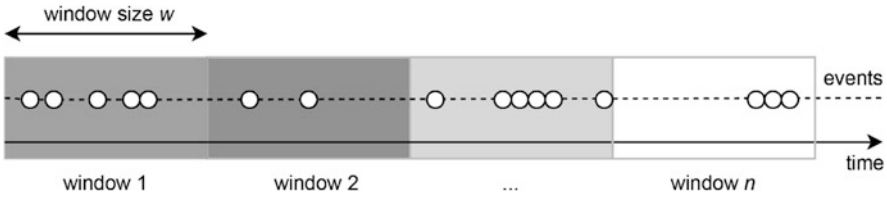


Fig. 2 Segmenting time into tumbling windows

aggregates for a multitude of features such as request intensity, header diversity, response code distribution, and insertion point diversity.

As depicted in Fig. 2, events from a source IP are grouped by dividing them into distinct tumbling windows of time that do not overlap. For each IP, we calculate features within each window across the IP's events to create the feature vectors used in the training process. These feature vectors form the individual samples that compose each app's dataset. This process is detailed further in Sects. 3.5 and 4.1.

The size of the tumbling window affects storage requirements, processing time, and computation of costs in production. Therefore, striking a balance between these factors and model performance should be considered in a real-world environment. A window size $w \in \{1, 30, 60, 300\}$, measured in seconds, is used to create different datasets to be compared and evaluated.

3.3 Data Labelling

The AppSensor events consist of all matches against the attack sensors and patterns, which are then aggregated into tumbling windows per source IP. However, each window needs to be labelled as to whether it belongs to a scanner or not. This is highly challenging to do by hand given the volume of windows. Therefore, the method of labelling the data is based on analysis and comparison between generated data and production data.

We generate data by deploying WebGoat [37], a deliberately insecure web app to demonstrate common server-side application flaws with many vulnerabilities that can be exploited, thus providing the coverage needed [29, 37]. WebGoat is then scanned with two DAST tools, OWASP Zed Attack Proxy [30] and PortSwigger Burp Scanner [18]. Both widely used scanners are included in the Kali Linux pen test platform, an open-source distro specifically for pen testing and security research [9]. The data analysis technique employed is EDA [33], used to understand the data, its shape, the types of features, and the relationship and patterns between them. The results are then used to verify intuitions about the characteristics of scanning.

The labelling of a source IP's aggregated events within a given window is determined by combining insights gained from the data analysis with manual classification by a human expert. This allows the creation of a set of rules to

label activity as originating from a DAST scanner or not. The WebGoat data were manually labelled by a human expert who had access to a dashboard containing the statistics for each source IP session, for example, the frequency distribution of sensor activation and rates of requests being made, plus the label predicted by the ruleset. The expert corrected the labels by hand where needed, and then the generalised rules were used to label real-world production data gathered from the AppSensor stream.

These rules were generated using known attack behaviours, anomaly detection thresholds, and expert consultations, and some are based on proprietary information.

3.4 Data Analysis

Salient input features to the ML model are key in allowing the classifier to detect DAST scanning activity. Given we are adopting a tumbling window approach that aggregates features, before deciding how to engineer these features, it is useful to analyse the data in combination with our DAST domain knowledge. Note that, where relevant in this subsection, the figures use logarithmic axes and are for a single app. We also find similar trends across the other apps in our dataset.

3.4.1 Unique URIs per Source IP

Figure 3 illustrates the empirical cumulative distribution function (eCDF) for the number of distinct URIs requested per IP. Scanning attacks that request only one distinct URI are likely to be single-page scans, where the URI stays constant and

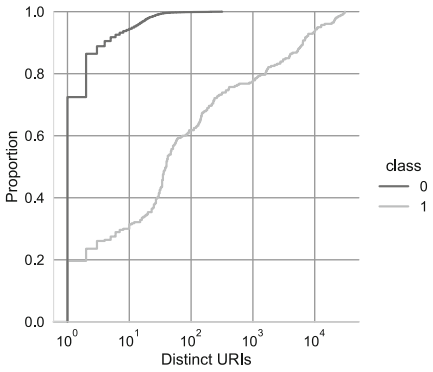


Fig. 3 eCDF for distinct URIs for `app-1`

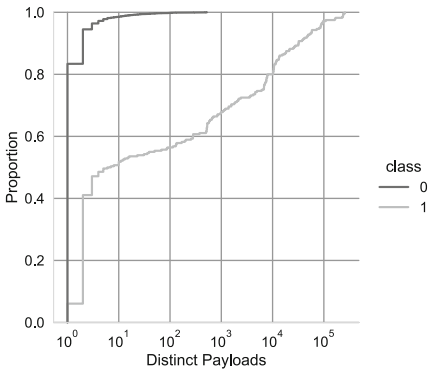


Fig. 4 eCDF for distinct payloads for `app-1`

the payload changes for each request. On the other hand, non-scanning attacks often have a large number of distinct URIs, likely because web crawlers tend to trigger the `s4xx` sensor rather than carrying a malicious payload. Given the clear distinction between attack and non-attack classes, counting the unique URIs within an aggregated tumbling window seems like a sensible approach.

3.4.2 Number of Payloads per Session

The eCDF for the number of distinct payloads per session is shown in Fig. 4. The significant proportion of attack sessions with only a small number of requests is likely due to rapid-fire requests made in a very short time. With the clear distinction between the attack and non-attack classes, we find it reasonable to count both unique and total payloads within a tumbling window.

3.4.3 HTTP Method Distribution Across Events

An interesting observation can be made in Fig. 5 where only the attack class has any matches for the HTTP methods `PROPFIND` and `TRACK`. These two methods can be abused for the injection and extraction of sensitive information [20, 21]. Given the differences in the plot, we decide that a count of each HTTP method in a tumbling window may be a useful feature.

3.4.4 Event Intensity per Class

In Fig. 6, a scatter plot is presented that displays the correlation between the intensity of events (per second) and the duration of each session. The upper right corner, which represents the longest and most intense sessions, is where the red points

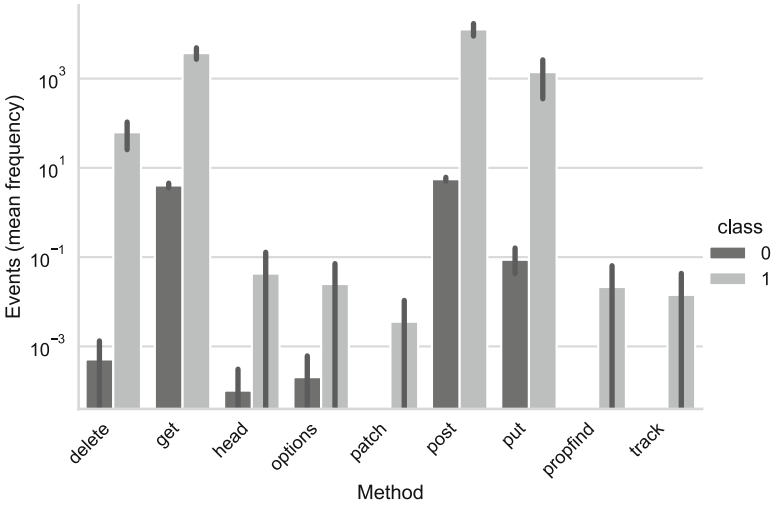
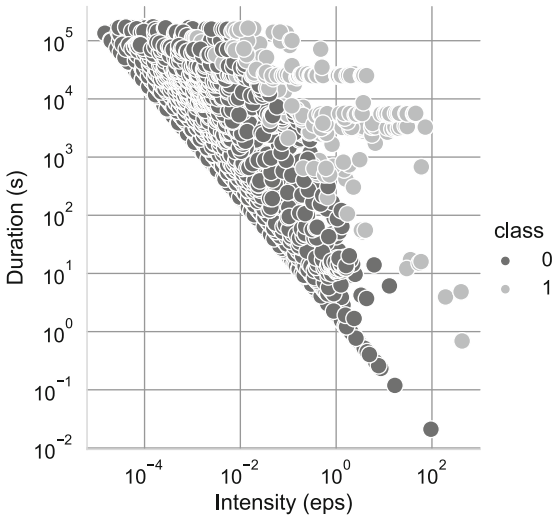


Fig. 5 Distribution of events per HTTP method for the analysed data, showcasing the mean frequency on a logarithmic scale, categorised by 'class'

Fig. 6 Intensity and duration distributions for `app-1`



indicative of the DAST scanner attack class are clustered. The lack of points in the lower left corner signifies the inverse relationship between duration and intensity. The data suggest that DAST attacks tend to last longer and are more intense, reinforcing the value of using aggregated time windows. This leads us to believe that a DAST attack will generate significantly more events than non-DAST activity, making aggregated features potentially more discriminating.

Table 3 Makeup of a single event

Feature	Description
`timestamp`	Event timestamp (e.g. `2023-05-12 15:30:45`)
`ip`	Source IP address (e.g. `192.168.0.1`)
`payload`	Extracted payload data (e.g. `<script>alert("XSS");</script>`, `' OR 1=1 -`)
`method`	HTTP method used (e.g. `GET`, `POST`)
`headers.known`	Known headers in the request (e.g. `User-Agent`, `Content-Type`)
`headers.unknown`	Unknown headers in the request (e.g. `X-Custom-Header`)
`sensor`	Type of sensor triggered (e.g. `xss`, `sqli`)
`pattern`	Specific behavior or pattern detected (e.g. `IFrame Tag Injection`, `SQL Comment Sequence`)
`location`	Location of the action or data (e.g. `Body`, `Header`, `Cookie`)
`uri`	URI of the event (e.g. https://www.example.com/api/endpoint)

3.5 Crafting the Features

In our dataset, each sample is an aggregation of individual events from a specific source IP within a set time frame. These are labelled as either a DAST attack or non-DAST activity. Our approach to representing these data is a two-step process: first, we need a representation for each distinct event; next, we aggregate these individual events pertaining to a specific source IP.

Table 3 outlines the features we extract from each event, and Table 4 describes how we combine these features to create a holistic representation of all events from a specific source IP. It is important to note that the source IP plays a crucial role in our aggregation process, but it does not make its way into the final feature vector.

For numerical features, we stick to conventional methods such as summation or counts, also applying min-max scaling [23]. On the other hand, the text features need a bit more finesse. It stands to reason that if we can capture and include unique text features that are typical of DAST attacks in our machine learning model, we could enhance our ability to identify these attacks. For instance, a DAST scanner typically generates a host of unique payloads, which could be a tell-tale sign of its activity.

For the textual data in our `payload`, `headers.known`, `headers.unknown`, and `URI` features, we calculate the TF-IDF score [19, 26] to rank their importance within the training data. We limit this to the top 100 terms, effectively creating a vocabulary of the most frequent and significant terms across a given corpus. The goal here is to highlight the unique strings that DAST scanners use while filtering out the more commonplace ones. Within a given time period, we can then calculate the count of these top 100 terms for each of the aforementioned features.

Table 4 Makeup of an aggregated window of events per source IP

Feature	Aggregation	Description
`ip`	N/A	Source IP (identifier, not used in model training)
`payload`	value counts	Occurrence count of top 100 payloads
	nunique	Number of unique payloads
`method`	value counts	Occurrence count of each method
`sensor`	value counts	Occurrence count of each sensor trigger
`pattern`	value counts	Occurrence count of each pattern trigger
`location`	value counts	Occurrence count of each location
`headers`	count	Total number of headers
`headers.known`	count	Number of known headers
	nunique	Number of unique known headers
	value counts	Occurrence count of top 100 known headers
`headers.unknown`	count	Number of unknown headers
	nunique	Number of unique unknown headers
	value counts	Occurrence count of top 100 unknown headers
`uri`	value counts	Occurrence count of top 100 URIs
	nunique	Number of unique URIs
Overall	N/A	Final vector of 469 elements, excluding source IP

3.6 Random Forest ML Model

We have decided to use the random forest algorithm for training our models, drawing inspiration from its successful application in areas such as network intrusion detection systems [1, 5, 13].

Each application in our scenario has its own dedicated model, with the window size w adjusted as necessary. When tested, the model's job is to make a binary prediction: Is the aggregated activity of a source IP within a time window indicative of a DAST attack or not?

The random forest algorithm is particularly suitable for this task because it is robust, adaptable, and effectively handles high-dimensional data. Moreover, it tends to perform well in mitigating overfitting, a common challenge in machine learning. This means that even when faced with new and unseen data, our model remains reliable in its predictions.

4 Setting Up the Experiment

Our experiment draws on data from three of the most active real-world web applications, which are protected by the NGWAF. These data were gathered over a period of 48 hours, providing us with a comprehensive and high-quality dataset from a broad range of sensors.

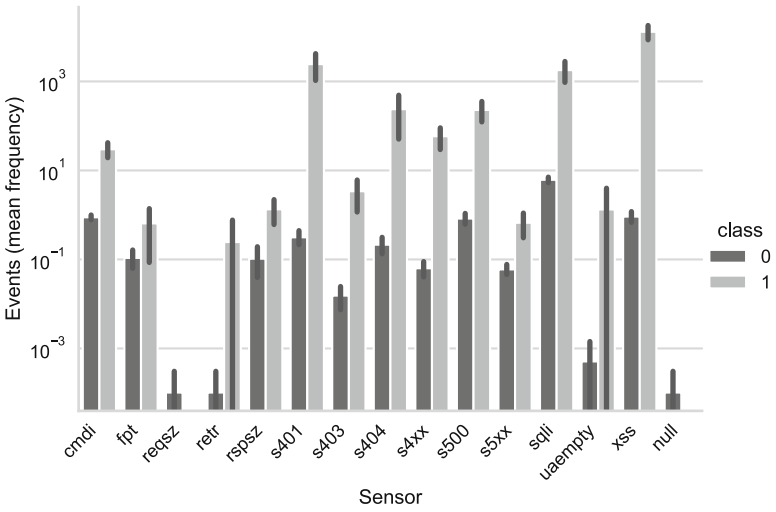


Fig. 7 Event distribution per sensor for `app-1`, displaying the mean frequency on a logarithmic scale, categorised by 'class'

4.1 The Dataset

In terms of volume, `app-1` contributed 49,033,476 events, `app-2` added 27,328,196 events, and `app-3` offered 18,113,350 events. As an illustration, Fig. 7 shows the event distribution for `app-1`. Here, you can see that certain types of attacks, such as SQLi and XSS, are markedly more common than others, reflected by the taller bars. It is worth noting that the y-axis is logarithmic, so the height difference between the bars is even more pronounced than it appears. This distribution confirms our expectation that DAST scanning attacks far outnumber non-scanning attacks in real-world scenarios.

We collected the events from all three apps simultaneously from the AppSensor stream. This approach ensures consistency in our definition of an attack, as the same set of detection patterns is applied to all events. We have withheld specific details about the apps' functionality and nature for reasons of confidentiality.

To ensure the integrity of our dataset, we also performed deduplication to eliminate duplicate events that might have been captured in the stream. By utilising our proprietary data, we have been able to design an experiment specifically tailored to AppSensor-orientated scenarios, which gives us greater confidence in our findings.

Table 5 Number of samples in train and test splits per window size w across each app

w	App	# train DAST	# train non-DAST	# test DAST	# test non-DAST
1	app-1	129,941	129,941	1,265,655	58,605
	app-2	183,333	183,333	409,687	78,334
	app-3	7704	7704	3287	3303
30	app-1	105,458	105,458	1,144,586	44,805
	app-2	135,662	135,662	244,617	58,200
	app-3	7714	7714	4177	3303
60	app-1	93,532	93,532	1,139,166	39,810
	app-2	133,283	133,283	230,000	57,042
	app-3	7707	7707	4187	3303
300	app-1	76,583	76,583	1,138,835	33,348
	app-2	128,479	128,479	155,857	55,110
	app-3	7704	7704	3287	3303

4.2 Splitting the Dataset

Table 5 provides an overview of the distribution of DAST and non-DAST samples in the training and testing datasets for each application. Remember that each sample represents a collection of activity from a specific source IP within a tumbling window and is tagged as either a DAST scanning attack or non-DAST activity. We have ensured an even distribution of each class in the training split by subsampling the DAST scan samples.

We sorted the data for each app chronologically and then divided it into a training split and a test split. After this division, we generated the tumbling windows. The training split comprises all the data collected in the initial 24 hours, while the test split includes all data from the subsequent 24-hour collection period. This approach presents a rigorous test scenario where a model is trained using data from the first day and then tested on data from the following day—mirroring a real-world deployment scenario.

> important

A vulnerability scan attack can last anywhere from a few minutes to a couple of hours, depending on the configuration. Therefore, random sampling events might create a train-test split that includes the same sessions. This overlap risks having a scan included in both the training and testing splits, which could falsely inflate performance. To avoid this, we generate each source IP's aggregations within a tumbling window across each $w \in \{1, 30, 60, 300\}$, following the steps detailed in Sect. 3.5, after splitting the event data.

4.3 Evaluating the Model

The performance of our machine learning model is evaluated based on four metrics: precision, recall, F1 score, and miss rate. While all of these metrics provide valuable information, we particularly emphasise the F1 score and the miss rate. The F1 score helps us balance the precision and recall of our model, and the miss rate indicates the percentage of scanning attack samples that the model mistakenly identifies as non-scanning activities.

In our model, a DAST scanning attack is considered the positive case, while non-DAST activity is treated as the negative case. The formal definitions of these metrics are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

$$Miss\ rate = 100 \cdot (1 - Recall) \tag{4}$$

Another critical tool we use to evaluate our model is the receiver operating characteristic (ROC) curve. This curve helps us visualise the performance of our classifier at various probability thresholds. The area under the ROC curve, known as the AUC-ROC, quantifies our model's ability to distinguish between DAST and non-DAST activity. In essence, a higher AUC indicates that our model is better able to distinguish between these two categories.

4.4 Hyperparameters for Random Forest Model

Our random forest model uses the default parameters, as described in [24], which provide a good balance between performance and general applicability across various datasets. These defaults are designed to prevent overfitting, simplify model usage, and ensure reasonable computational efficiency.

Furthermore, we used the following software tools for our analysis: Jupyter [10], Python 3.9 [35], scikit-learn [16], pandas [15], and numpy [6].

5 Analysis of Experimental Results

We trained and evaluated a total of twelve models, three for each window size $w \in 1, 30, 60, 300$. Each model was trained using its corresponding training split and tested with its respective test split, as outlined in Table 5. This approach of having fixed train and test data, segmented into two separate 24-hour periods, ensures that any change in model performance can be directly attributed to the variation in w.

As shown in Table 6, our models show strong performance in all window sizes. The model with $w = 60$ performs the best in terms of detection of DAST activity, boasting an average F1 score of 0.94 and a miss rate of 6% in all three applications. However, for a production environment, $w = 30$ could also be a feasible choice, and it is plausible to consider varying w in smaller increments, such as every 5 seconds, for further tuning. Despite the potential for additional tuning, our results are highly encouraging, even with the smallest window size of $w = 1$, which could be particularly useful in a resource-constrained environment.

5.1 Performance Analysis via ROC Curve

The model's performance can be further demonstrated through the receiver operating characteristic (ROC) curve. For example, the ROC curve for `app-1` with $w = 60$, shown in Fig. 8, illustrates the trade-off between the true-positive rate (the proportion of DAST attacks correctly predicted) and the false-positive rate (the

Table 6 Performance metrics per w across each app

w	App	Precision	Recall	F1	Miss rate
1	app-1	0.97	0.95	0.96	5%
	app-2	0.89	0.82	0.84	18%
	app-3	0.96	0.96	0.96	4%
	Average	0.94	0.91	0.92	9%
30	app-1	0.98	0.96	0.97	4%
	app-2	0.92	0.89	0.89	11%
	app-3	0.96	0.96	0.96	4%
	Average	0.95	0.94	0.94	6.33%
60	**app-1**	**0.98**	**0.97**	**0.97**	**3%**
	app-2	0.92	0.89	0.90	11%
	app-3	0.96	0.96	0.96	4%
	Average	**0.95**	**0.94**	**0.94**	**6%**
300	app-1	0.97	0.85	0.90	15%
	app-2	0.90	0.86	0.87	14%
	app-3	0.96	0.96	0.96	4%
	Average	0.94	0.89	0.91	11%

The bold values are the maximum values.

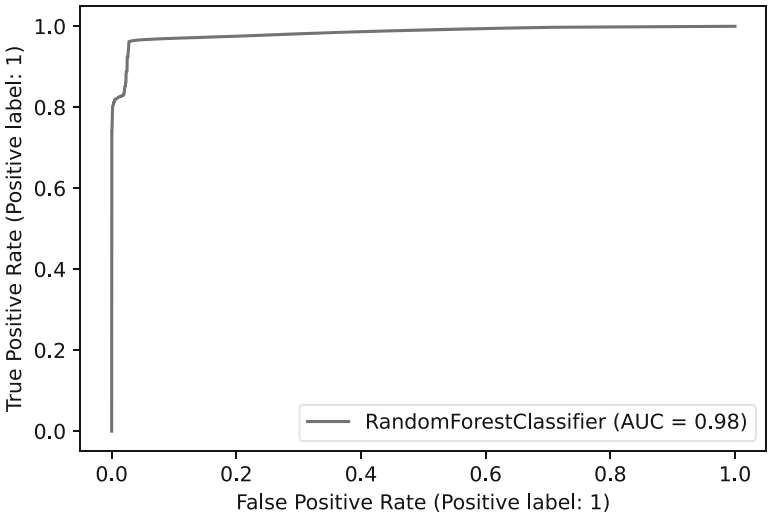


Fig. 8 ROC curve of the model for `app-1` with $w = 60$

proportion of DAST attack activity mistakenly predicted as non-DAST). Ideally, we aim for a point at the top left corner of the ROC space, which signifies a true-positive rate of 1 and a false positive rate of 0. The steepness of the ROC curve is key to maximising the true-positive rate while minimising the false-positive rate. A high area under the curve (AUC) implies a high-performance model capable of accurately predicting both positive and negative samples. Indeed, the AUC of 0.98 for `app-1`, as shown in Fig. 8, suggests that the model is close to optimal performance.

5.1.1 Discussion on Feature Engineering and Future Directions

As indicated in Sect. 3.4, DAST attacks are typically of longer duration, with distinct payloads, URIs, and a larger volume of events. We believe that our careful feature engineering, which captures these aspects, played a significant role in achieving the impressive results. In a practical setting, an additional automated step could be incorporated into the NGWAF to block source IPs associated with DAST activity, eliminating the need for manual intervention. This would make our proposed end-to-end system highly beneficial in a production environment.

However, we acknowledge that models need to be updated over time. This is a relatively manageable task that involves collecting new data and retraining the model periodically, perhaps weekly. Notwithstanding, our use of a high-quality proprietary dataset for these three selected apps instils strong confidence in our machine learning model's capability to detect DAST scanning attacks, especially those perpetrated by unskilled individuals operating from a single IP.

6 Conclusions

In this chapter, we presented a random forest machine learning model designed to detect DAST vulnerability scanning attacks, acting as an extension to an existing NGWAF. Our solution can be integrated into an end-to-end streaming data pipeline, allowing the classification and subsequent blocking of DAST activity. Using a large, real-world dataset containing millions of events, our experiments revealed that an optimal window size of 60 seconds leads to an F1 score of 0.94 and an impressively low miss rate of 6% on average in three selected enterprise-grade production applications.

Future work could explore the development of a suite of machine learning models, one for each sensor, and investigate the potential of sequence-orientated deep learning techniques, such as transformers, recurrent networks, and convolutions.

References

1. Alqahtani, H., Sarker, I.H., Kalim, A., Minhaz Hossain, S.M., Ikhlaq, S., Hossain, S.: Cyber intrusion detection using machine learning classification techniques. In: Computing Science, Communication and Security, pp. 121–131. Springer Singapore, Singapore (2020)
2. Choraś, M., Kozik, R.: Machine learning techniques applied to detect cyber attacks on web applications. Logic J. IGPL **23**(1), 45–56 (2014). https://doi.org/10.1093/jigpal/jzu038
3. Denning, D.E.: An intrusion-detection model. IEEE Trans. Softw. Eng. **2**, 222–232 (1987)
4. European Union Agency For Cybersecurity: ENISA threat landscape 2020 - web application attacks. Tech. rep., ENISA (2021). https://www.enisa.europa.eu/publications/web-application-attacks
5. Farnaaz, N., Jabbar, M.: Random forest modeling for network intrusion detection system. Procedia Comput. Sci. **89**, 213–217 (2016)
6. Harris, C.R., Millman, K.J., van der Walt, S.J., et al.: Array programming with NumPy. Nature **585**(7825), 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2
7. Hyslip, T.S.: Cybercrime-as-a-Service Operations, pp. 815–846. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-319-78440-3_36
8. Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. EAI Endorsed Trans. Secur. Safety **3**(9), e2 (2016)
9. Kali Linux: Kali tools: Kali linux tools. https://www.kali.org/tools/ (2023)
10. Kluyver, T., Ragan-Kelley, B., Pérez, F., et al.: Jupyter notebooks – a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (eds.) Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87–90. IOS Press, Amsterdam (2016)
11. Kruegel, C., Vigna, G.: An anomaly detection of web-based attacks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 251–261 (2003)
12. Millar, S., Podgurskii, D., Kuykendall, D., Martínez del Rincón, J., Miller, P.: Optimising vulnerability triage in dast with deep learning. In: Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security, AISec'22, pp. 137–147. Association for Computing Machinery, New York, NY, USA (2022)
13. Negandhi, P., Trivedi, Y., Mangrulkar, R.: Intrusion detection system using random forest on the NSL-KDD dataset. In: Emerging Research in Computing, Information, Communication and Applications, pp. 519–531. Springer, New York (2019)

14. Pan, Y., Sun, F., Teng, Z., et al.: Detecting web attacks with end-to-end deep learning. J. Internet Serv. Appl. **10**(1), 1–22 (2019)
15. Pandas: Pandas-dev/pandas: Pandas (2023). https://doi.org/10.5281/zenodo.3509134
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., et al.: Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
17. Perez-Villegas, A., Torrano-Gimenez, C., Alvarez, G.: Applying markov chains to web intrusion detection. In: Proceedings of Reunión Espanola sobre Criptología y Seguridad de la Información (RECSI 2010), pp. 361–366 (2010)
18. Portswigger: BURP scanner – web vulnerability scanner from portswigger. https://portswigger.net/burp/vulnerability-scanner (2023)
19. Qaiser, S., Ali, R.: Text mining: use of tf-idf to examine the relevance of words to documents. Int. J. Comput. Appl. **181**(1), 25–29 (2018)
20. Rapid7: HTTP track. https://www.rapid7.com/db/vulnerabilities/http-track-method-enabled/ (2023)
21. Rapid7: WebDAV propfind method allows web directory browsing. https://www.rapid7.com/db/vulnerabilities/http-generic-propfind-dir-browsing/ (2023)
22. Saha, A., Sanyal, S.: Application layer intrusion detection with combination of explicit-rule-based and machine learning algorithms and deployment in cyber-defence program. CoRR **abs/1411.3089** (2014). http://arxiv.org/abs/1411.3089
23. Scikit-learn: Scikit-learn: Preprocessing Min-Max Scaler. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html (2023)
24. Scikit-learn: Scikit-learn: Random Forest Classifier. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (2023)
25. Shone, N., Ngoc, T.N., Phai, V.D., He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Trans. Emerg. Top. Comput. Intell. **2**(1), 41–50 (2018)
26. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Document. **28**(1), 11–21 (1972)
27. Sun, F., Zhang, P., White, J., Schmidt, D., Staples, J., Krause, L.: A feasibility study of autonomically detecting in-process cyber-attacks. In: 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), pp. 1–8. IEEE (2017)
28. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 dataset. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. IEEE (2009)
29. The OWASP Foundation: OWASP Top Ten. https://owasp.org/www-project-top-ten/ (2023)
30. The OWASP Foundation: OWASP Zed Attack Proxy (ZAP). https://www.zaproxy.org/ (2023)
31. The OWASP Foundation: Project AppSensor. https://owasp.org/www-project-appsensor/ (2023)
32. Torrano-Giménez, C., Perez-Villegas, A., Alvarez Maranón, G.: An anomaly-based approach for intrusion detection in web traffic. J. Inf. Assur. Secur. **5**(4), 446–454
33. Tukey, J.W., et al.: Exploratory Data Analysis, vol. 2. Addison-Wesley, Reading, MA (1977)
34. UCI Machine Learning Repository: UCI machine learning repository: Kdd cup 1999 dataset. https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data (2023)
35. Van Rossum, G., Drake Jr, F.L.: Python Reference Manual. Centrum voor Wiskunde en Informatica, Amsterdam (1995)
36. Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 41525–41550 (2019)
37. WebGoat: WebGoat 8: A deliberately insecure web application. https://github.com/WebGoat/WebGoat (2023)
38. Yin, C., Zhu, Y., Fei, J., He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access **5**, 21954–21961 (2017)

Enhancing Embedded IoT Systems for Intrusion Detection Using a Hybrid Model

Mansour Alqarni and Akramul Azim

1 Introduction

Intrusion detection for IoT networks is a critical task that involves the use of various techniques and tools to monitor network traffic and detect potential security breaches or attacks in real-time. This is particularly important in the context of IoT networks, which typically comprise a large number of interconnected devices with varying levels of security and vulnerability. With the digitization of more and more devices, users are learning to leverage the power of the Internet to improve their day-to-day lives with the help of wireless signals and automation. However, the ubiquity of the Internet has facilitated commerce, social engagement, and other forms of digital communication to such a degree that the connectivity of any new device is almost taken for granted or simply assumed. These new devices typically become part of a vast network of other devices that has been popularly termed, the "Internet of Things" (IoT). The IoT is a collection of different kinds of devices that are connected through a network, can be controlled by the Internet, and can interact with one another easily. However, due to a lack of proper security, devices are vulnerable to different kinds of cyberattacks because of their continuous exposure to the largely unsecured network that is the Internet. Adversaries could scan the network for vulnerable devices or applications on the network. Cloud computing has made the IoT more available and also stores different kinds of data from the IoT

M. Alqarni (✉)
Fanshawe College, London, ON, Canada

Ontario Tech University, Oshawa, ON, Canada
e-mail: mansour.alqarni@ontariotechu.net

A. Azim
Ontario Tech University, Oshawa, ON, Canada
e-mail: akramul.azim@ontariotechu.ca

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

T. Sipola et al. (eds.), *Artificial Intelligence for Security*,
https://doi.org/10.1007/978-3-031-57452-8_15

network. While digital security is certainly not an unattainable goal, the principal concern is not any one device, but is instead the large volume of devices that are connecting to the IoT every day. In addition, these new devices are precipitating changes in how people use the Internet, making it very difficult for security experts to ward off cyberattacks [20].

The study of intrusion detection for IoT networks aims to mitigate these security threats by identifying potential vulnerabilities in IoT networks and developing effective solutions to detect and prevent attacks in real-time. With the development of advanced techniques such as deep learning neural networks and improved datasets, the accuracy and efficiency of intrusion detection for IoT networks have improved significantly. Therefore, it is essential to continue research and development in this field to ensure the security and privacy of IoT networks and their users.

The rise in popularity of cryptocurrency mining has provided opportunities for attackers to exploit other devices, networks, and even electricity to carry out their mining operations. As a result, this has led to a surge in intrusion attacks, creating a range of new pathways and vulnerabilities for cybercriminals to exploit. However, traditional intrusion detection systems (IDS) are not always well-suited to the unique characteristics of IoT networks. With low-powered devices and limited computing resources and often operating in remote or inaccessible environments, IoT networks require specialized techniques for intrusion detection. To address this issue, researchers have proposed a range of specialized techniques for intrusion detection in IoT networks, including machine learning algorithms, anomaly detection, and lightweight cryptographic protocols. Machine learning algorithms are capable of learning and identifying normal behavior patterns within the network and detecting any deviations from those patterns that may indicate a potential intrusion. Anomaly detection is another technique that can identify unusual traffic patterns and behavior that could indicate an intrusion. Lightweight cryptographic protocols are also used to protect the network from unauthorized access, by ensuring secure communication between devices.

Overall, as IoT networks continue to grow and expand, it is crucial to have effective intrusion detection techniques in place to safeguard against potential attacks. By utilizing specialized techniques such as machine learning algorithms, anomaly detection, and lightweight cryptographic protocols, it is possible to provide a high level of security and protect against the increasing threat of intrusion attacks in IoT networks.

The key contributions of this paper are outlined as follows:

1. Dataset Enhancement: In the initial phase, we conducted a comprehensive analysis of the dataset, focusing on addressing the data imbalance issue. Recognizing the potential bias introduced by imbalanced data, we diligently undertook measures to ensure an equitable and balanced distribution. This strategic balancing of the dataset is pivotal in ameliorating the accuracy of intrusion detection, as it curtails the skewed influence of dominant classes.
2. Hybrid DAIDS-RNN Model: Subsequently, we introduce a novel and robust hybrid intrusion detection model by seamlessly integrating the deep

autoencoder-based intrusion detection system (DAIDS) with recurrent neural networks (RNNs). This novel amalgamation harnesses the proficiency of DAIDS in capturing intricate spatial patterns and the temporal sensitivity of RNNs in delineating sequential dependencies. Our model employs the latent representation generated by DAIDS alongside the temporal insights provided by the RNN component. This synergy not only enriches the feature representation but also equips the model with a holistic understanding of network dynamics over time, thus enhancing its efficacy in identifying anomalies.

3. Hyperparameter Optimization: In the final phase, we embrace a meticulous hyperparameter optimization strategy to fine-tune the deep RNN architecture. This optimization procedure systematically explores the parameter space, thereby maximizing the model's discriminatory power and accuracy in detecting intrusions. By adapting the hyperparameters to the intricacies of the hybrid DAIDS-RNN model, we aim to achieve optimal performance.

Through the amalgamation of these contributions, including the hybrid DAIDS-RNN model, dataset balancing, and hyperparameter tuning, we endeavor to substantially elevate intrusion detection accuracy. Our empirical assessment substantiates the potency of our approach, underscoring marked advancements in intrusion detection precision. This methodology holds substantial potential in reinforcing IoT device security and safeguarding sensitive information in a digitally interconnected landscape.

The subsequent sections of the paper are structured as follows:

- Section 2: Comprehensive Overview of Related Work
- Section 3: Dataset Description, Collection, and Preprocessing
- Section 4: Hybrid DAIDS-RNN Model for Intrusion Detection
- Section 5: Experimental Results and Configurations
- Section 6: Conclusion

Each section is meticulously crafted to delineate the theoretical foundation, experimental design, and results of our hybrid DAIDS-RNN intrusion detection model.

2 Comprehensive Overview of Related Work

2.1 Related Work

Intrusion detection in the context of Internet of Things (IoT) networks is a multifaceted challenge, characterized by the need to safeguard interconnected devices from a diverse range of cyber threats. Understanding the landscape of existing research is crucial for advancing effective intrusion detection methodologies tailored to IoT environments. This section offers a comprehensive overview of the related work, providing insights into the diverse strategies that researchers have

employed to address the intricate security concerns in IoT networks. Boumkheld et al. [9] employed a conventional machine learning approach coupled with a naive Bayesian network to evaluate the algorithm's potential in intrusion detection. In a similar vein, Jokar et al. [21] introduced Zigbee-based Q-learning as a strategy to bolster network security against intrusion, reporting favorable outcomes in terms of attack monitoring. Hasan et al. [18] proposed a hybrid architecture, combining a convolutional neural network (CNN) and long short-term memory (LSTM), for classifying electricity information attributes. Meanwhile, Wang et al. [39] advocated a hierarchical approach for selecting pertinent features from intrusion detection networks.

The combination of CNN and LSTM algorithms has been utilized effectively for intrusion detection [21].

By examining and synthesizing prior research endeavors, we gain valuable insights into the efficacy, limitations, and unexplored avenues within the field [3, 29, 37]. This collective understanding paves the way for the introduction of our innovative hybrid intrusion detection algorithm, which combines the strengths of deep autoencoder-based intrusion detection system (DAIDS) and recurrent neural networks (RNNs). As we delve into the various intrusion detection techniques, it is important to note that the field encompasses a spectrum of approaches ranging from machine learning-based anomaly detection to signature-based methods. The complexity of IoT networks necessitates an adaptive and nuanced response, leading to the emergence of hybrid models that harmonize multiple methodologies for enhanced accuracy and robustness.

Given that IoT devices have the ability to collect a significant amount of data, it is important to leverage deep learning methods for detecting various kinds of intrusion attacks [30]. Building, implementing, and deploying IoT platforms involve considering security as one of the most important factors. The Internet of Things (IoT) pertains to the capability of communicating with, monitoring, and operating automated objects through the Internet. However, IoT is vulnerable to a variety of cyberattacks due to its limited processing capabilities, low power, and restricted technology. Consequently, security measures like cryptography and authentication are challenging to implement. An intrusion detection system is a deep learning model that can be trained to detect different kinds of intrusions in a network. In contrast, cloud application firewalls are not capable of providing complete security to a network since they fail to recognize and skip new types of attacks. Intrusion detection systems, on the other hand, continuously monitor the abnormalities of network data to detect any intrusions [34].

Intrusion detection can be classified into two categories: misuse detection and anomaly detection. A misuse detection mechanism is primarily designed to detect the misuse of processor, RAM, and storage of devices that can be controlled by attackers to dictate the functionality of a system. An anomaly detection system, on the other hand, records the system's normal behavior and monitors its behavior. Any aberrant system behavior is registered as an attack. It is crucial to implement intrusion detection systems for IoT networks since traditional intrusion detection systems are not well-suited to IoT networks' unique characteristics.

These networks have low-powered devices and limited computing resources and often operate in remote or inaccessible environments. Therefore, researchers have proposed specialized techniques for intrusion detection in IoT networks, including machine learning algorithms, anomaly detection, and lightweight cryptographic protocols [2, 12, 15, 23].

Machine learning algorithms and deep learning algorithms, for example, have become quite popular strategies for different kinds of vulnerability and intrusion detection mechanisms. A research group proposed an approach that involved a six-layer neural network that was designed for intrusion detection [24]. The researchers used the NSL-KDD dataset [36] to facilitate this project. Another group of researchers used a binary bat algorithm, and they detected an intrusion attack by finding local minima [14]. There are some popular datasets for supervised learning-based algorithms, and previous researchers also used these datasets for intrusion detection. Some researchers have used artificial neural networks and have claimed 84% accuracy [17]. Other researchers have used machine learning-based algorithms [28, 33] on different datasets and achieved good accuracy. Earlier, in 2005, some researchers used support vector machine algorithms, decision tree classifiers, and GA-based feature selection for their intrusion detection efforts [11, 31]. Santo et al. proposed an information gain-based approach for intrusion detection [22]. The proposed system is divided into two stages. First, the features are extracted from the dataset. Then, in the second stage, the SVM classifier is used to classify the network data. Recently, some researchers used deep learning algorithms [4] and achieved good experimental results.

The research put forth by other researchers working in the field has proven to be very significant insofar as their results have highlighted the importance of the learning model in the construction of a mechanism that is responsive to increasingly sophisticated threats. The advantage of a deep learning model is the simple fact that it is designed to not only assess the particulars of a given scenario but also respond and develop or evolve over time. With cyberattackers constantly seeking to improve the sophistication of their intrusion protocols, deep learning models must be able to react to isolated incursion but then also draw a lesson from that particular encounter, one that can be leveraged to address or thwart future attacks.

2.2 Existing Intrusion Detection Techniques

The domain of IoT intrusion detection encompasses an array of techniques aimed at safeguarding the integrity and security of interconnected devices. In this section, we present a comprehensive survey of existing intrusion detection methodologies, highlighting their approaches, advantages, and limitations. This exploration underscores the need for innovative solutions that can address the evolving landscape of IoT threats. Anomaly detection approaches [26] Anomaly-based intrusion detection techniques focus on identifying deviations from established behavioral patterns. Traditional statistical methods, such as mean-variance analysis and clustering, have

been adapted to IoT environments. These methods, while effective for certain scenarios, can struggle to capture complex nonlinear relationships and the dynamic nature of IoT traffic. Wenjuan et al. [25] used signature-based detection involves creating a database of known attack patterns and identifying matches within network traffic. While efficient in recognizing known attacks, this method is limited by its inability to detect novel or zero-day threats. Furthermore, the sheer diversity of IoT devices and communication protocols challenges the feasibility of maintaining comprehensive signature databases. Other researchers [1, 35] did their experiments with machine learning-based techniques, including supervised and unsupervised methods, have gained prominence in IoT intrusion detection. Algorithms such as decision trees, support vector machines, and k-nearest neighbors have been adapted to classify network traffic as normal or malicious. However, the effectiveness of these methods hinges on the availability of labeled training data, which can be scarce and subject to class imbalances. In some advanced papers such as [13, 19, 27] addressing the temporal aspect of network traffic, time-series analysis techniques have been explored. These methods, including hidden Markov models and recurrent neural networks (RNNs), focus on capturing sequential dependencies in data. RNNs, in particular, excel in modeling temporal relationships but might not fully harness the spatial patterns present in network traffic.

While each of these techniques contributes to the understanding and detection of IoT intrusions, they also exhibit limitations when applied in isolation. The dynamic and diverse nature of IoT networks necessitates a holistic approach that can capture both spatial intricacies and temporal dynamics. In the subsequent sections, we delve into the incorporation of these insights into our hybrid DAIDS-RNN model, aiming to bridge the gaps left by stand-alone methodologies.

By presenting this comprehensive panorama of existing intrusion detection techniques, we lay the foundation for our innovation—leveraging hybridity and synergy to develop an advanced approach that surpasses the limitations of individual methods. Our model's ability to harmoniously integrate spatial and temporal insights marks a crucial step toward more effective and adaptable IoT intrusion detection. Therefore, it is the aim of this study to address the aforementioned gaps in the research.

3 Dataset Description and Preprocessing

3.1 Dataset Overview

The IoT-23 dataset [32] includes network traffic data from 23 different IoT devices, capturing both benign and malicious activities. It's specifically designed for evaluating intrusion detection algorithms in IoT networks. The IoT-23 dataset serves as the foundational corpus for our research, facilitating the evaluation and validation of our hybrid DAIDS-RNN model for intrusion detection in Internet of Things (IoT) networks. This section offers a comprehensive overview of the dataset's

key characteristics, composition, and relevance to our investigation. The dataset is meticulously designed to mirror real-world IoT network traffic, encompassing a rich spectrum of benign and malicious activities. Its composition embodies the diversity of IoT devices and communication protocols, making it a fitting choice for assessing intrusion detection mechanisms tailored for IoT environments. The dataset comprises an extensive collection of network traffic instances, capturing interactions from 23 distinct IoT devices. These devices span a wide array of functionalities and encompass various device categories, ensuring a representative sample of IoT network behaviors. IoT-23 simulates a range of activities, encompassing legitimate communications and diverse attack scenarios that IoT networks can potentially encounter. The dataset's comprehensive coverage of both benign and malicious activities positions it as a robust substrate for evaluating the efficacy of intrusion detection models. The main goal we selected is IoT-23 dataset because it is specifically designed for evaluating intrusion detection in IoT environments. Autoencoders could be effective in capturing the complex and diverse patterns of network traffic generated by different IoT devices.

3.2 Data Preprocessing

The reliability and effectiveness of any machine learning model heavily depend on the quality of the input data. In this section, we outline the meticulous preprocessing steps undertaken to transform the raw IoT-23 dataset into a format conducive to training and evaluating our hybrid DAIDS-RNN model for intrusion detection. We have used both an existing dataset, and we collected our dataset from a real-world IoT system that contains different kinds of information with attack types. The dataset is carefully designed to capture various types of attacks commonly encountered in IoT networks, such as denial of service (DoS), distributed denial of service (DDoS), malware infections, and unauthorized access attempts. These attack instances are included to provide a realistic representation of the security threats faced by IoT devices. To ensure the dataset's quality and reliability, extensive data preprocessing techniques have been applied. These techniques include data cleaning, removal of redundant or irrelevant features, and normalization of data values. Additionally, to address the issue of class imbalance, the dataset has been carefully balanced, ensuring that both normal and attack instances are adequately represented. Figure 1 shows some portion of our dataset. **Data Cleaning and Filtering**: We eliminated any instances that contained incomplete or erroneous information that could adversely affect the learning process of our model. **Feature Normalization**: To ensure uniformity across features and prevent any attribute from disproportionately influencing the model, we applied feature normalization. This process scaled all features to a common range, facilitating effective learning across the entire dataset. **Feature Extraction**: We employed feature extraction techniques to convert the raw network traffic data into meaningful features that would contribute to the performance of our model. This process involved identifying

dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	label
0.00	0.00	0.05	0.00	normal
0.00	0.00	0.00	0.00	normal
1.00	1.00	0.00	0.00	neptune
0.03	0.01	0.00	0.01	normal
...	...	...	...	...
0.01	0.00	0.00	0.00	normal
0.00	0.00	0.07	0.07	back
0.00	0.00	0.00	0.00	normal
0.00	0.00	0.44	1.00	mscan

Fig. 1 Sample of intrusion detection dataset

key attributes and aggregating them into features that encode relevant patterns (Fig. 2). Figure 3 shows the distribution of different kinds of attacks and normal events. **Data Splitting**: The dataset was divided into distinct subsets for training, validation, and testing purposes. This partitioning ensured that the model's performance was assessed on unseen data, thus providing a more accurate evaluation of its generalization capabilities. **Label Encoding**: To facilitate supervised learning, we encoded the labels for each instance—differentiating between normal behavior and various attack categories—into a format suitable for our model's training process; see Fig. 1.

It seems that there is a huge imbalance between the normal and attack data. Figure 4 shows the final class count for attack and normal events. Typically it refers to a significant disparity in the number of instances between classes. While a 5% difference may not seem substantial at first glance, it can still impact the performance of machine learning models, especially in cases where one class is underrepresented. Here are some reasons why a 5% difference might be considered significant. **Impact on model learning** in situations where one class is significantly smaller than the other: machine learning models may struggle to learn patterns associated with the minority class. This can result in models that are biased toward the majority class and perform poorly on the minority class. **Imbalanced costs** in some applications: the cost of misclassifying certain instances can be high. Even a small difference in the number of instances between classes can lead to imbalanced costs if the minority class contains critical cases. For instance, as a network does not have any control mechanisms to limit traffic to only legitimate access, any attacker could attempt to use brute force or gain access to remote control services. In such scenarios, having a comprehensive and balanced dataset for intrusion detection becomes crucial. Even a minor imbalance in the dataset can impact the ability to detect and mitigate these threats effectively.

Since the result indicate that can the dataset is not balanced, undersampling methods [7] have been applied to balance the dataset. It is important to remove the dataset imbalance because that can produce biased results after training and testing. Figure 5 shows the balanced dataset.

The number of normal events and attack events are the same in within the balanced dataset. This balanced dataset provides more accurate results than an imbalanced dataset.

Attack Type	Parameters Type	Services
Normal	protocol_type	Whois, Private, HTTP
	dst_bytes	
	src_bytes	
Land	wrong_fragment	Telnet
Neptune	num_failed_logins	Telne, HTTP, FTP, UUCP
	logged_in	
	num_compromised	
	root_shell	
	su_attempted	
	num_root	
	num_file_creations	
	num_shells	
	num_access_files	
	num_outbound_cmds	
	is_host_login	
	is_guest_login	
Back	srv_count	HTTP
	serror_rate	
	srv_serror_rate	
	rerror_rate	
	srv_rerror_rate	
	same_srv_rate	
	diff_srv_rate	
	srv_diff_host_rate	
	dst_host_count	
	dst_host_srv_count	
Mscan	dst_host_same_srv_rate	ecr_i
	dst_host_diff_srv_rate	
	dst_host_same_src_port_rate	
	dst_host_srv_diff_host_rate	
	dst_host_serror_rate	
	dst_host_srv_serror_rate	

Fig. 2 Example of our dataset and parameters (Features)

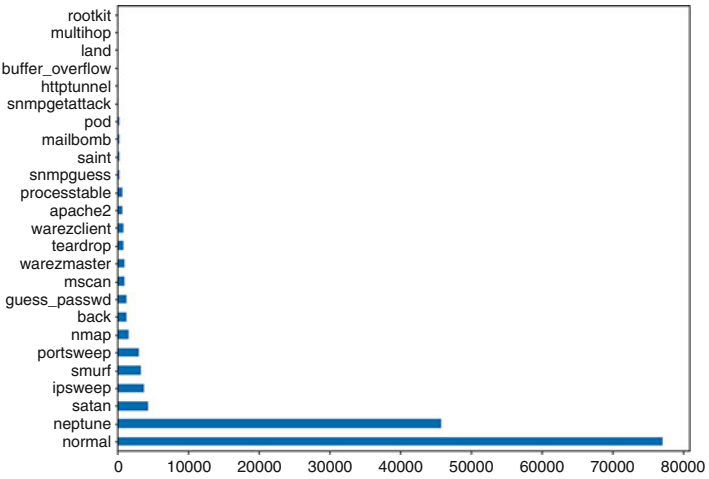


Fig. 3 Distribution of different attacks and normal event

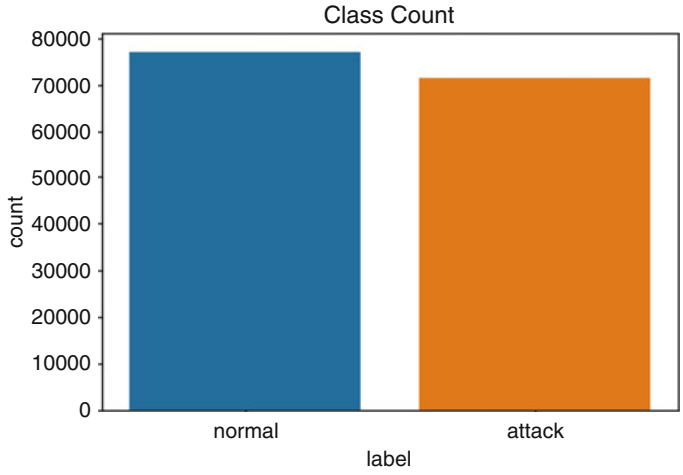


Fig. 4 Class count for normal and attack

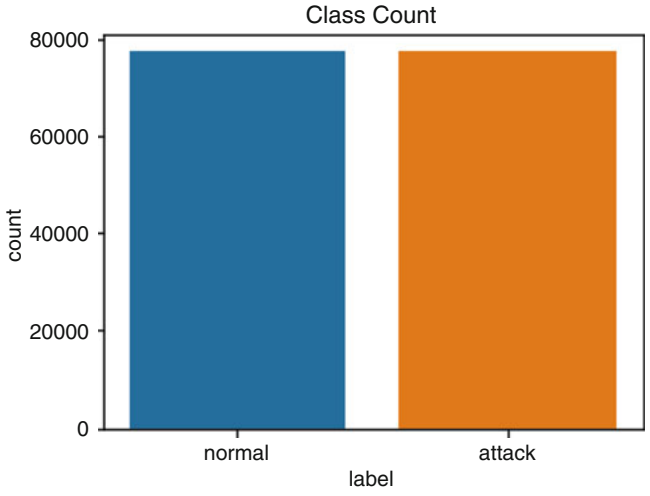


Fig. 5 Intrusion detection balanced dataset

3.3 Dataset Combination for Enhanced Performance

In pursuit of a comprehensive and robust intrusion detection model for IoT networks, we have taken a deliberate step toward amalgamating the strengths of two prominent datasets: the "IoT Network Intrusion Dataset" [38] and the "IoT-23 dataset." [32] This strategic fusion not only bolsters the dataset's size but also enriches the diversity of network behaviors captured, thereby enhancing the efficacy of our hybrid DAIDS-RNN model for intrusion detection.

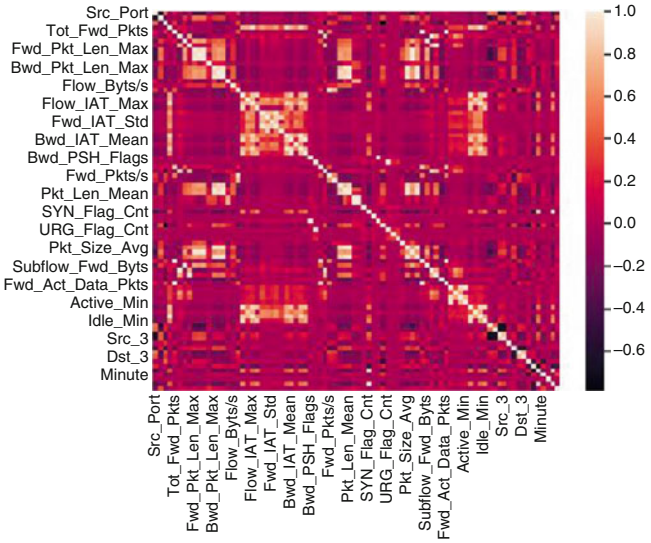


Fig. 6 Visualization of correlations between the features

3.4 Data Imbalance

We have also removed the data imbalance problem [10, 16] and made the dataset balanced. We have chosen to use a combined dataset "IoT Network Intrusion Dataset" [38]. The reason we choose this dataset for this project is that it has the common attacks we could predict for IoT network. However, the dataset requires substantial amount of preprocessing and contains a huge class imbalance data. The other thing we have to look for is the correlation of each column with every other column. If they are perfectly correlated, then it means one of them completely defines the other; thus it contains no new information and can be safely removed. In Fig. 6 we illustrated a sample of the data visualization technique using heatmap. The correlation looks a bit conjugated. We filter out the ones that have perfect correlation result shown in Fig. 7, after we removed the second column from each pair of correlated columns to remove the duplicated data.

4 Hybrid DAIDS-RNN Model for Intrusion Detection

This paper proposes a state-of-the-art model for IoT intrusion detection by leveraging recurrent neural networks (RNN) with an attention mechanism. The RNN model with attention effectively captures sequential dependencies in network traffic data and focuses on the most relevant parts of the sequence for intrusion detection.

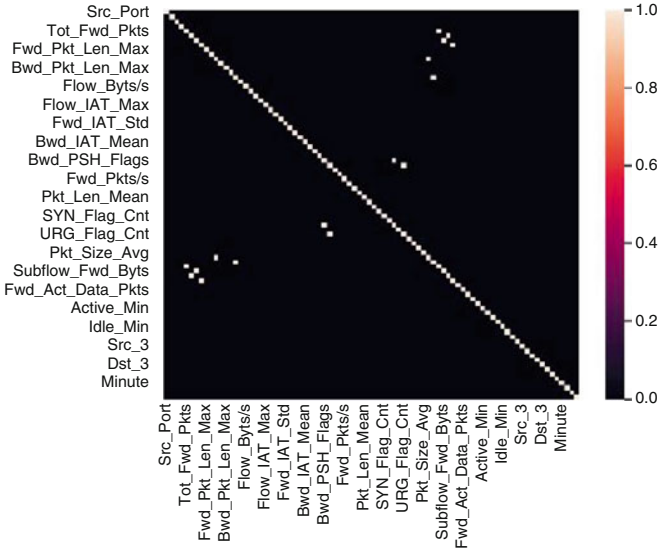


Fig. 7 Instances of perfect correlation shown in white

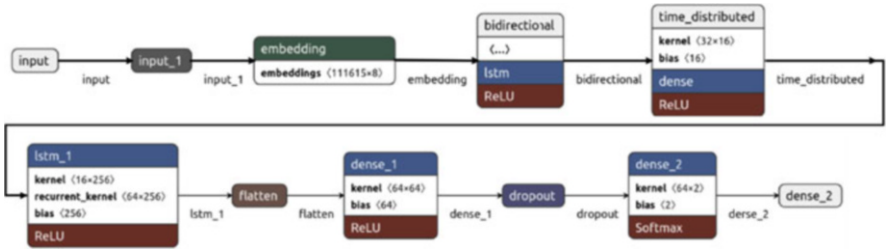


Fig. 8 Hybrid DAIDS-LSTM model

In this paper, we present the architecture of the model, its working principles, and its performance evaluation using a mixed IoT datasets [6].

Our hybrid DAIDS-RNN model represents an innovative integration of the deep autoencoder-based intrusion detection system (DAIDS) with the power of recurrent neural networks (RNNs), more specifically the long short-term memory (LSTM) architecture. Figure 8 shows the hybrid DAIDS-LSTM model architecture. This carefully crafted fusion allows us to tackle the nuanced challenges posed by intrusion detection in dynamic Internet of Things (IoT) networks, leveraging both spatial and temporal insights for enhanced accuracy.

The model architecture comprises the following components.

4.1 Input Layer

The input layer of our hybrid DAIDS-LSTM model serves as the initial point of interaction between the raw network traffic data and the subsequent layers of feature extraction and analysis. This section outlines the preprocessing steps applied to the input data to ensure its compatibility with the model's architecture.

4.2 DAIDS Component

The DAIDS (deep autoencoder-based intrusion detection system) component lies at the heart of our innovative hybrid DAIDS-LSTM model, working synergistically with the LSTM-based RNN to provide a multidimensional approach to intrusion detection. The DAIDS component focuses on capturing spatial patterns within network traffic data, while the subsequent LSTM component delves into temporal dynamics. The DAIDS component serves as the spatial feature extractor, meticulously capturing underlying spatial patterns within the network traffic data. This is achieved through a deep autoencoder architecture, which encodes the raw input data into a compressed latent representation. The encoding process can be mathematically described by the following equations:

Encoder:

$$h_1 = \sigma(W_2 x + b_2)$$

$$h_2 = \sigma(W_2 h_1 + b_2)$$

Latent representation:

$$z = \sigma\ (W_3 h_2 + b_3)$$

4.2.1 Encoder Architecture

The encoder within the DAIDS component is a neural network that transforms the input data through a series of layers, each contributing to the gradual extraction of spatial features.

Layer 1

- Activation Function: ReLU (rectified linear unit)
- Transformation: $h_1 = \sigma(W_2 x + b_2)$

Description: The first layer applies a linear transformation of the input data x using weights W_1 and biases b_1, followed by the ReLU activation function. This introduces nonlinearity and captures basic spatial characteristics.

Layer 2

- Activation Function: ReLU
- Transformation: $h_2 = \sigma(W_2 h_1 + b_2)$
  Description: The second layer processes the output of the first layer h_1 in a similar manner, further extracting complex spatial features.

Latent Representation Generation

The output of the encoder's final layer, h_2, is a high-dimensional representation that encapsulates spatial attributes. This representation, often referred to as the latent representation z, is a compressed and distilled version of the input data.

Layer 3

- Activation Function: ReLU
- Transformation: $z = \sigma\ (W_3 h_2 + b_3)$
  Description: The final layer transforms h_2 into the latent representation z, which encodes essential spatial features. This latent representation serves as a spatial fingerprint of normal behavior within the network traffic data.

Spatial Features Extraction

The latent representation z generated by the DAIDS component captures spatial patterns that define normal network behavior. Spatial features extraction acts as a spatial analyzer that distills critical spatial patterns from the network traffic data. This spatial insight is later integrated with temporal insights from the LSTM component, creating a comprehensive representation for intrusion detection in the hybrid DAIDS-LSTM model.

Significance

- z encodes spatial attributes such as device interactions, communication patterns, and typical data flows.
- Deviations from this spatial fingerprint could signify anomalous behavior or intrusion attempts.

The latent representation z encapsulates crucial spatial attributes that define normal network behavior. It forms a foundational component for our model's holistic understanding of network dynamics. The outcome of the encoder is the latent representation z, which serves as a compact yet comprehensive portrayal of the spatial attributes within the network traffic data. This latent representation captures the essence of normal behavior and serves as a reference point against which anomalies are evaluated.

4.3 Integration with LSTM

The integration between the DAIDS (deep autoencoder-based intrusion detection system) component and the LSTM (long short-term memory) layer constitutes a key innovation in our hybrid DAIDS-LSTM model. This integration harnesses the strengths of both spatial and temporal insights, thereby enhancing the model's capability for accurate intrusion detection. The integration process can be summarized as follows:

$$\text{LSTM}(\text{DAIDS}(x))$$

Where:

- x represents the raw network traffic data.
- DAIDS(x) signifies the transformation of x through the DAIDS component to produce the latent representation z.
- LSTM (.) indicates the application of the LSTM layer to the output of the DAIDS component. The result is an integrated output that captures both spatial and temporal insights. This equation succinctly conveys the process of integrating the DAIDS spatial insight with the LSTM temporal analysis to create a comprehensive representation for intrusion detection.

4.4 Hyperparameter Optimization

In this section, we outline the process of fine-tuning the hyperparameters of the hybrid DAIDS-LSTM model to optimize its performance for intrusion detection. **Hyperparameters Considered**: We carefully selected a range of hyperparameters that are pivotal in shaping the behavior and efficacy of our model. Table 1 shows the training model parameters of our deep learning model.

- **Learning Rate:** This hyperparameter dictates the step size in the parameter space during gradient descent. We experimented with values ranging from 0.001 to 0.1 to strike a balance between fast convergence and avoiding overshooting.

Table 1 Deep learning training model parameters

Parameters name	Values
Learning rate	0.001
Batch size	128
Epoch	150

- **Batch Size:** The number of samples used in each training iteration profoundly affects memory utilization and convergence speed. We explored batch sizes of 32, 64, and 128 to observe the effects on both training stability and computational efficiency.
- **Number of LSTM Units:** The number of LSTM units in the LSTM layer influences the model's capacity to capture temporal dependencies. We varied this hyperparameter between 128, 256, and 512 units to examine the trade-off between model complexity and generalization.
- **Dropout Rate:** Dropout is a regularization technique that prevents overfitting by randomly deactivating units during training. We experimented with dropout rates of 0.2, 0.5, and 0.8 to find the optimal level of regularization for our specific model architecture.
- **Number of Epochs:** The number of epochs determines how many times the model iterates through the entire dataset. We explored epochs ranging from 50 to 150, ensuring sufficient training time without overfitting.

It is very difficult, misleading, error-prone, and time-consuming to set the model parameters by tuning manually. Therefore, we have used hyperparameter tuning algorithms like random search [8] to find the best parameters easily. With a limited amount of processing effort, random search identifies the parameters at random and finds better models by effectively investigating a larger, less appealing configuration space.

Through this rigorous process of hyperparameter optimization, we ensured that the hybrid DAIDS-LSTM model is configured optimally to achieve exceptional intrusion detection accuracy in IoT environments. The chosen hyperparameters reflect a balance between model complexity, training efficiency, and robust performance.

Figure 9 shows the hyperparameter tuning architecture and process that we have used to find the best parameters for our proposed deep learning models.

5 Our Experiments and Analysis

In this section, we present a comprehensive evaluation of the performance of our proposed hybrid DAIDS-LSTM model for IoT intrusion detection. We compare the accuracy and effectiveness of our model against baseline methods to demonstrate its superior intrusion detection capabilities.

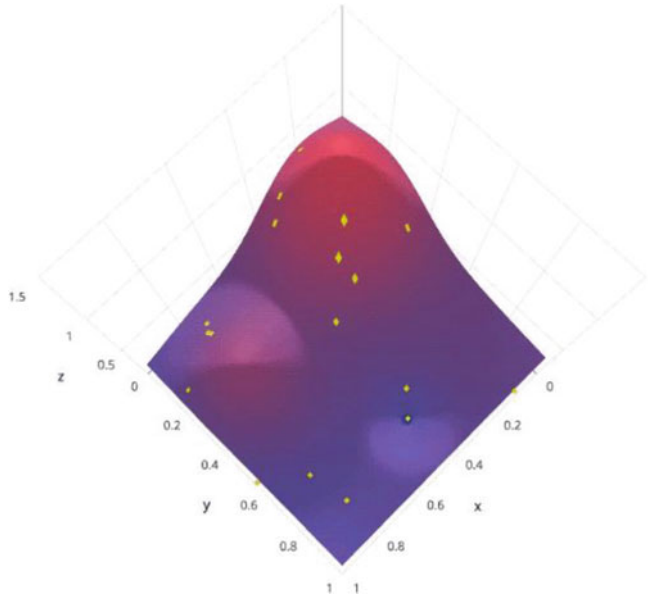


Fig. 9 Hyperparameter tuning architecture [8]

5.1 Environment We Created for this Experiment

This approach we used for our experimental research environment was the best approach to allow for normalized training parameters to improve both time and speed for our experiments. To achieve that, we used an Intel core i9 processor with Tesla k80 GPU and 16 GB of RAM; with the higher configuration, we could reject anything beyond the scope of unexpected behavior. To assess the effectiveness of our proposed model, we conducted experiments using a real-world IoT dataset. The dataset consists of network traffic data collected from various IoT devices in a controlled environment. We randomly split the dataset into training and testing sets, with a ratio of 80:20. The chosen ratio strikes a balance between providing enough data for training the model effectively and reserving a sufficient amount for reliable testing and evaluation.

5.2 Training, Validation, and Testing

This study has primarily focused on analyzing training, validation, and testing outcomes. To ensure rigorous evaluation, the dataset has been meticulously partitioned into three distinct subsets: the training dataset, the validation dataset, and the testing dataset.

Table 2 Training, validation, and testing result analysis from combined dataset

Training	Validation	Testing
99.85%	99.70%	99.64%

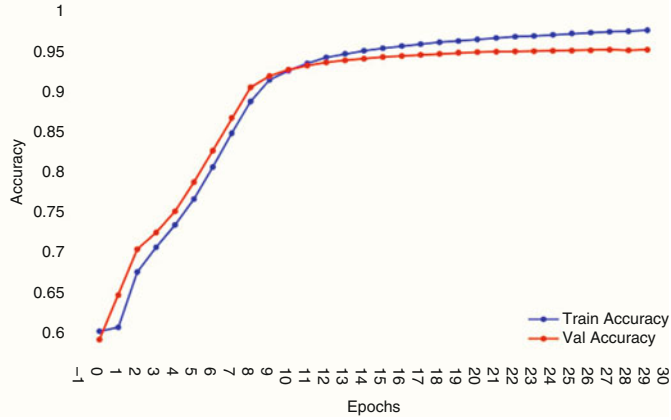


Fig. 10 Training and validation accuracy curve

The division of the dataset adheres to the following specifications: the training and validation datasets are split in an 80:20 ratio, with 80% of the data dedicated to training and 20% for validation purposes. Notably, a separate and entirely distinct testing dataset has been employed, encompassing the same number of instances as the validation dataset.

This approach guarantees that the model's performance is comprehensively assessed on unseen data, validating its ability to generalize beyond the training and validation phases. Such a meticulous dataset partitioning strategy forms a fundamental aspect of our experimental methodology, bolstering the credibility and reliability of the study's findings. **Result analysis:** Table 2 shows the training, validation, and testing result accuracy of our detection system.

Figure 10 shows the training vs validation accuracy result of the IoT intrusion detection system.

5.3 Evaluation Metric

We employed a range of standard evaluation metrics to comprehensively assess the performance of our model and compare it with baseline methods:

- Accuracy: The proportion of correctly classified instances among all instances.
- Precision: The ratio of correctly predicted positive instances to the total predicted positive instances.

Table 3 Comparison of results with previous research

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
CNN	96.60%	0.85	0.82	0.84	0.90
LSTM-CNN	98.80%	0.86	0.85	0.87	0.92
LSTM	98.95%	0.93	0.91	0.92	0.96
Hybrid DAIDS-LSTM	99.64%	0.96	0.97	0.96	0.99

- Recall: The ratio of correctly predicted positive instances to the total actual positive instances.
- F1-Score: The harmonic mean of precision and recall, providing a balanced measure. F-score = (2 * precision * recall)/(precision + recall)
- Area Under the ROC Curve (AUC-ROC): A measure of the model's ability to distinguish between normal and intrusive instances.

5.4 Results and Comparison

We present the results of our experimentation in a tabulated format, comparing the performance of our hybrid DAIDS-LSTM model with baseline methods. We tested deep learning models of previous researchers [4]. The metrics were calculated using a rigorous tenfold cross-validation strategy to ensure unbiased assessment. Table 3 shows the comparison of this study's result with previous researchers.

6 Conclusion

this study has introduced and extensively evaluated the hybrid DAIDS-LSTM model as an innovative and effective approach to IoT intrusion detection. This model harnesses the power of both spatial insights, extracted through the DAIDS component, and temporal dynamics, captured by the LSTM component, to achieve superior intrusion detection accuracy. The experimental results underscore the effectiveness of our proposed hybrid DAIDS-LSTM model in IoT intrusion detection. Its ability to seamlessly integrate spatial and temporal dimensions sets it apart from traditional approaches, resulting in remarkable accuracy and robustness across various intrusion scenarios. We made this dataset available [5] to encourage other researchers to continue pursuing all the necessary refinements that may improve the accuracy of intrusion detection mechanisms. Looking ahead, we plan to continue exploring ways to enhance intrusion detection in IoT systems. One promising avenue for future research is the utilization of advanced machine learning techniques, such as GPT-4, which has shown great potential in natural language processing and other domains. By incorporating these techniques into our model,

we aim to further improve the accuracy and efficiency of our intrusion detection system. In conclusion, further enhancements and refinements to the hybrid DAIDS-LSTM model are anticipated, propelling it to the forefront of IoT security. This work encourages continued exploration in the realm of hybrid models, blending spatial and temporal insights, and their applications in emerging security challenges. Also, we believe that our findings will contribute to the ongoing effort to enhance the security of IoT systems.

References

1. Aboelwafa, M.M.N., Seddik, K.G., Eldefrawy, M.H., Gadallah, Y., Gidlund, M.: A machine-learning-based technique for false data injection attacks detection in industrial IoT. IEEE Internet Things J. **7**(9), 8462–8471 (2020). https://doi.org/10.1109/JIOT.2020.2991693
2. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. J. Netw. Comput. Appl. **60**, 19–31 (2016)
3. Ahmed, S.W., Kientz, F., Kashef, R.: A modified transformer neural network (MTNN) for robust intrusion detection in IoT networks. In: 2023 International Telecommunications Conference (ITC-Egypt), pp. 663–668 (2023). https://doi.org/10.1109/ITC-Egypt58155.2023.10206134
4. Alkahtani, H., Aldhyani, T.H.: Intrusion detection system to advance Internet of Things infrastructure-based deep learning algorithms. Complexity **2021** (2021)
5. Alqarni, M.: IoT intrusion detection dataset (2022). https://www.kaggle.com/datasets/malqarni/iotdataset
6. Alqarni, M., Azim, A.: Software source code vulnerability detection using advanced deep convolutional neural network. In: Proceedings of the 31st Annual International Conference on Computer Science and Software Engineering, pp. 226–231 (2021)
7. Bach, M., Werner, A., Palt, M.: The proposal of undersampling method for learning from imbalanced datasets. Proc. Comput. Sci. **159**, 125–134 (2019)
8. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(2) (2012)
9. Boumkheld, N., Ghogho, M., El Koutbi, M.: Intrusion detection system for the detection of blackhole attacks in a smart grid. In: 2016 4th International Symposium on Computational and Business Intelligence (ISCBI), pp. 108–111. IEEE, Piscataway (2016)
10. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: 2010 20th International Conference on Pattern Recognition, pp. 3121–3124. IEEE, Piscataway (2010)
11. Cleetus, N., Dhanya, K.: Multi-objective functions in particle swarm optimization for intrusion detection. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 387–392. IEEE, Piscataway (2014)
12. da Costa, K.A., Papa, J.P., Lisboa, C.O., Munoz, R., de Albuquerque, V.H.C.: Internet of Things: a survey on machine learning-based intrusion detection approaches. Comput. Netw. **151**, 147–157 (2019). https://doi.org/https://doi.org/10.1016/j.comnet.2019.01.023. https://www.sciencedirect.com/science/article/pii/S1389128618308739
13. Djellali, C., Adda, M.: A new hybrid deep learning model based-recommender system using artificial neural network and hidden Markov model. Procedia Computer Science **175**, 214–220 (2020). https://doi.org/https://doi.org/10.1016/j.procs.2020.07.032. https://www.sciencedirect.com/science/article/pii/S1877050920317129. The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology

14. Enache, A.C., Sgârciu, V.: A feature selection approach implemented with the binary bat algorithm applied for intrusion detection. In: 2015 38th International Conference on Telecommunications and Signal Processing (TSP), pp. 11–15 (2015). https://doi.org/10.1109/TSP.2015.7296215
15. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. **28**(1–2), 18–28 (2009)
16. Grzymala-Busse, J.W., Stefanowski, J., Wilk, S.: A comparison of two approaches to data mining from imbalanced data. J. Intell. Manuf. **16**(6), 565–573 (2005)
17. Hanif, S., Ilyas, T., Zeeshan, M.: Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset. In: 2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT and AI (HONET-ICT), pp. 152–156. IEEE, Piscataway (2019)
18. Hasan, M.N., Toma, R.N., Nahid, A.A., Islam, M.M., Kim, J.M.: Electricity theft detection in smart grid systems: a CNN-LSTM based approach. Energies **12**(17), 3310 (2019)
19. Hashish, I.A., Forni, F., Andreotti, G., Facchinetti, T., Darjani, S.: A hybrid model for bitcoin prices prediction using hidden Markov models and optimized LSTM networks. In: 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 721–728 (2019). https://doi.org/10.1109/ETFA.2019.8869094
20. Iasiello, E.: Cyber attack: A dull tool to shape foreign policy. In: 2013 5th International Conference on Cyber Conflict (CYCON 2013), pp. 1–18. IEEE, Piscataway (2013)
21. Jokar, P., Leung, V.C.: Intrusion detection and prevention for ZigBee-based home area networks in smart grids. IEEE Trans. Smart Grid **9**(3), 1800–1811 (2016)
22. Kareem, S., SS, K., Mostafa, R., RR, M.: An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection, vol. 2022. Basel (2022)
23. Khan, M.N., Rao, A., Camtepe, S.: Lightweight cryptographic protocols for IoT-constrained devices: a survey. IEEE Internet Things J. **8**(6), 4132–4156 (2021). https://doi.org/10.1109/JIOT.2020.3026493
24. Kiran, K.S., Devisetty, R.K., Kalyan, N.P., Mukundini, K., Karthi, R.: Building a intrusion detection system for IoT environment using machine learning techniques. Proc. Comput. Sci. **171**, 2372–2379 (2020)
25. Li, W., Tug, S., Meng, W., Wang, Y.: Designing collaborative blockchained signature-based intrusion detection in IoT environments. Fut. Gener. Comput. Syst. **96**, 481–489 (2019). https://doi.org/https://doi.org/10.1016/j.future.2019.02.064. https://www.sciencedirect.com/science/article/pii/S0167739X18327237
26. Lin, X.X., Lin, P., Yeh, E.H.: Anomaly detection/prediction for the internet of things: state of the art and the future. IEEE Netw. **35**(1), 212–218 (2021). https://doi.org/10.1109/MNET.001.1800552
27. Martins, A., Mateus, B., Fonseca, I., Farinha, J.T., Rodrigues, J., Mendes, M., Cardoso, A.M.: Predicting the health status of a pulp press based on deep neural networks and hidden Markov models. Energies **16**(6) (2023). https://doi.org/10.3390/en16062651. https://www.mdpi.com/1996-1073/16/6/2651
28. Mohammed, A., Saleh, A., Majdi, B., Muder, A., Salwani, A.: Intrusion detection for IoT based on a hybrid shuffled shepherd optimization algorithm. In: The Journal of Supercomputing. The Journal of Supercomputing (2022)
29. Muhammad, K., Hussain, T., Tanveer, M., Sannino, G., de Albuquerque, V.H.C.: Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks. IEEE Internet Things J. **7**(5), 4455–4463 (2020). https://doi.org/10.1109/JIOT.2019.2950469
30. Saeed, A., Ahmadinia, A., Javed, A., Larijani, H.: Intelligent intrusion detection in low power IoTs. ACM Trans. Internet Technol. **16**(4), 1–25 (2016)
31. Saxena, H., Richariya, V.: Intrusion detection in kdd99 dataset using SVM-PSO and feature reduction with information gain. Int. J. Comput. Appl. **98**(6) (2014)

32. Sebastian Garcia Agustin Parmisano, M.J.E.: Iot-23: A labeled dataset with malicious and benign IoT network traffic (version 1.0.0) [data set]. Zenodo (2020). http://doi.org/10.5281/zenodo.4743746
33. da Silva, A.S., Wickboldt, J.A., Granville, L.Z., Schaeffer-Filho, A.: Atlantic: a framework for anomaly traffic detection, classification, and mitigation in SDN. In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, pp. 27–35. IEEE, Piscataway (2016)
34. Suresh, G.M., Madhavu, M.L.: Ai based intrusion detection system using self-adaptive energy efficient bat algorithm for software defined IoT networks. In: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6. IEEE, Piscataway (2020)
35. Tahsien, S.M., Karimipour, H., Spachos, P.: Machine learning based solutions for security of internet of things (Iot): a survey. J. Netw. Comput. Appl. **161**, 102630 (2020). https://doi.org/https://doi.org/10.1016/j.jnca.2020.102630. https://www.sciencedirect.com/science/article/pii/S1084804520301041
36. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6 (2009). https://doi.org/10.1109/CISDA.2009.5356528
37. Thant, Y.M., Su Thwin, M.M., Htwe, C.S.: Iot network intrusion detection using long short-term memory recurrent neural network. In: 2023 IEEE Conference on Computer Applications (ICCA), pp. 334–339 (2023). https://doi.org/10.1109/ICCA51723.2023.10182005
38. Ullah, I., Mahmoud, Q.H.: A scheme for generating a dataset for anomalous activity detection in IoT networks. In: Canadian Conference on Artificial Intelligence, pp. 508–520. Springer, Berlin (2020)
39. Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., Zhu, M.: HAST-IDS: learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE Access **6**, 1792–1806 (2017)