Non-Compliance as Insecurity:
from Compliance Verification to Attack Derivation

PhD Thesis

Gianpietro Castiglione

UNIVERSITÀ
degli STUDI
di CATANIA

Department of Mathematics and Computer Science
PhD in Computer Science

PhD Thesis

Non-Compliance as Insecurity: from Compliance Verification to Attack Derivation

Gianpietro Castiglione

2022 - 2025

PhD Thesis Information

Candidate:	Gianpietro Castiglione
Supervisor:	Prof. Giampaolo Bella, Università degli Studi di Catania
PhD Coordinator:	Prof. Dario Catalano, Università degli Studi di Catania
Reviewers:	Prof. Paolo Balboni, Maastricht University Prof. Roberto Di Pietro, King Abdullah University of Science and Technology
PhD Cycle:	XXXVIII
University:	Università degli Studi di Catania

Declaration of Authorship

By signing this, I affirm that this Doctoral Thesis is original with no submissions for any other title. When information is taken from other sources, credit is given. This Doctoral Thesis draws inspiration from academic articles developed during the PhD and, in the best cases, presented at conferences, published in journals, or submitted. As allowed by the individual publications, the text presented herein may contain information that is the same as or strikingly similar to the original articles. The writer is the primary author and/or contributor of the included works, and all sources are referenced.

Gianpietro Castiglione

November 2025

Ethical and Legal Note

This Doctoral Thesis contains the detailed description and analysis of methodologies utilised for deriving and composing attacks, leveraging the status of *non-compliance*. However, all the content described herein is presented for the sole purpose of academic research and scientific study. Any application on real systems must be appropriately compliant with applicable laws. The ethical and guiding principle is strictly related to promoting risk awareness. Therefore, the Author disclaims all responsibility for any misuse, illegal, or unethical application of the information and methodologies described in this document by third parties.

In the space within my heart.

– Massive Attack

Abstract

Security threats and their consequences for institutions, businesses, and society are widely acknowledged. This well-established awareness has surely contributed to the birth of numerous instruments for the common regulation of the security domain. Worldwide institutions, especially the European ones, enact complex security legislation to protect the citizens and the infrastructure. This legislation, which encompasses regulations and directives, aims to safeguard data and harmonise security across the European region. Addressed entities must navigate this evolving legal landscape to implement and update their security measures.

However, although the legislative text serves to provide a common overview of security practices, its implementation is far from being easily adopted to cover the relevant technical aspects. It follows that understanding, implementing and correlating this legislation towards full compliance and technical security can be difficult and expensive.

This dissertation proposes a research stream aimed at deconstructing security legislation, that is, identifying weaknesses in its underlying structure for uncovering potential shortcuts in *compliance verification* and *attack derivation*. These aspects are examined by designing sequential methodologies and developing concrete tools that, particularly focusing on the NIS 2 Directive, starting with an analysis of the legislative text, culminate in vulnerability assessment.

First, ontologies are used to model the security measures provided in the Directive in a structured manner, allowing the relationships between legal concepts, requirements and controls to be represented in a formalised and structured way. Ontological reasoning then allows compliance to be verified, identifying ambiguities, discrepancies or gaps between regulatory obligations and practical implementation. Attack patterns relating to the security measures of the directive are subsequently

identified. The combination of semantic methods, such as semantic similarity and ontological semantics, appropriately codified in algorithms and human expertise, allows the detection of potentially exploitable attack patterns, highlighting possible weaknesses derivable from non-compliance with the measures themselves. These patterns form the basis for building killchains, in which different security frameworks are correlated to simulate realistic attack chains, showing how the attack patterns found can be used in sequence for offensive security activities. Finally, vulnerability confirmation is deepened by advancing an integrated technique with machine learning and fuzzing, allowing the presence of any non-compliance-derived software vulnerability to be confirmed.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Framing the Dissertation

As far as we can discern, the sole purpose of human existence is to kindle a light in the darkness of mere being.

— Carl Jung

1.1 What Is The Cyberthreat Landscape?

Living in the infosphere [51] has the side effect of exposing individuals and societies to conflicts that are increasingly recurring within it. It is a historical fact that we live in the age of cyberwarfare, where the insecurity of devices, systems, and networks is more and more exploited for structured, distributed, and wide-ranging attacks. The scope of these attacks now extends to the level of entire nations, targeting digital systems embedded in national territories.

This phenomenon is increasingly the result of existing geopolitical tensions, which further aggravate the risks and consequences of cyber conflicts impacting worldwide nations because of their support for the belligerents. As Europeans, a historical perspective on the relations between conventional warfare and cyber operations can begin with the start of the Russian offensive in Ukraine in 2022, resulting in a critical case study in the evolution of state-sponsored cyber operations. A recently published report by the Italian TIM group [132] analysed the cyberthreat scenario from 2022 to 2025, showing how military conflicts have emphasised the rise of cyberconflicts. As Figure 1.1 illustrates, cyberattack incidents (in particular DDoS ones) exhibited a moderate trend throughout 2022 and the first half of 2023 before beginning to spike in August, when the first spike was noted. A second rise is recorded between November 2023 and January 2024, during a period when the conflict between Israel and Hamas is erupting and the Russian-Ukrainian conflict is becoming more intense.

In addition, ENISA notes a rise in DDoS occurrences, which doubled in frequency from July 2023 to June 2024 when compared to all events recorded, rising from 21% to 41% of all events.

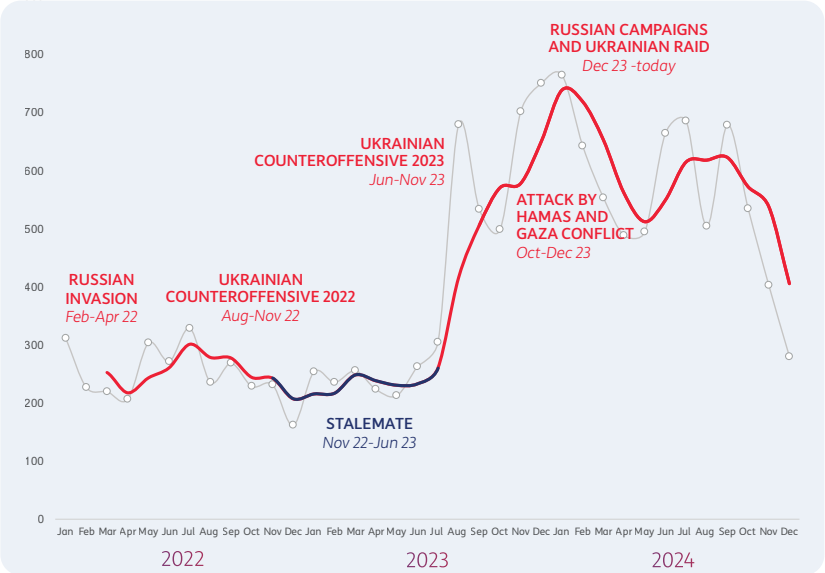


Figure 1.1: Relation between military conflicts and cyber conflicts (period 2022-2024) [132]

According to a Google report [58], there was a 250% spike in 2022 that targeted users in Ukraine and a more than 300% increase that targeted users in NATO countries, compared to a baseline set in 2020. Attackers backed by the Russian government carried out each of these attacks. Phishing is still a popular first step for attackers with official backing. In order to serve Russian national interests, attackers use this access to carry out certain Russian strategic objectives, such as obtaining intelligence, erasing data, and disclosing information.

Between 2021 and 2022, phishing efforts were made against specific targets by government-backed attackers (Figure 1.2). In 2022, Russian government-backed attackers targeted more Ukrainian users than users in any other country. As part of their pre-existing focus on Ukraine, several attackers (FROZENBARENTS, FROZENLAKE) stepped up their targeting of key infrastructure, utilities, and public services, as well as the media and information arena.

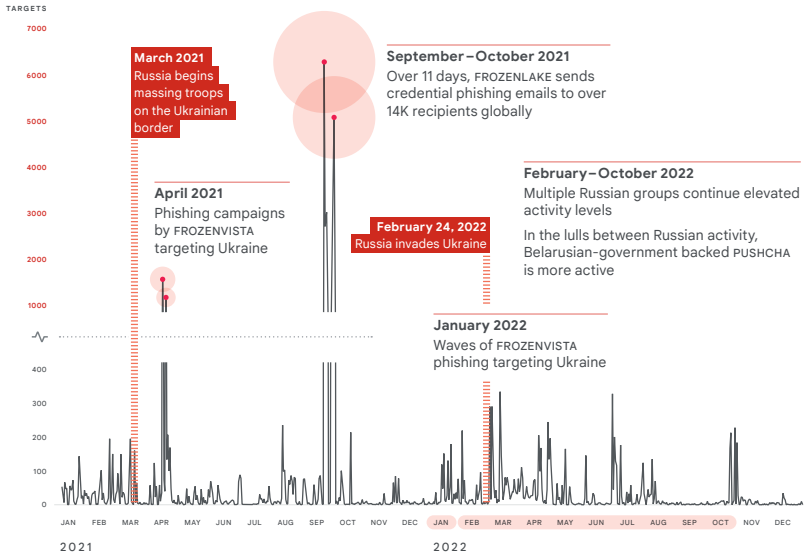


Figure 1.2: Fog of War — How the Ukraine Conflict Transformed the Cyber Threat Landscape [58]

Contextually to this geopolitical scenario, nations not directly involved in the armed conflict can also still be subject to cyber attacks, due to their political or logistical support to the belligerent states. In Italy, while the weight of low-intensity events (about 87%) was very high in 2019 and the years that followed, the incidence of high-intensity events, which now account for nearly 40% of the total, has tended to increase in recent years, as Figure 1.3 demonstrates. In 2024, a polarisation can be observed into extreme classes, focusing on those with varying degrees of intensity.

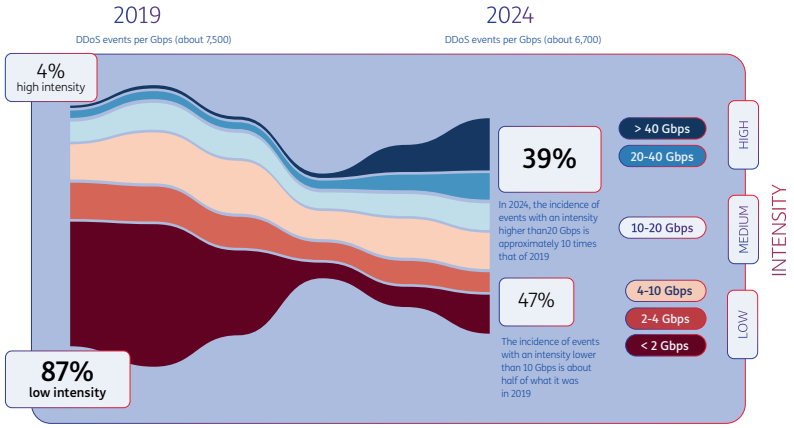


Figure 1.3: Intensity of DDoS events in Italy (period 2019-2024) [132]

A thorough analysis of academic literature and technical reports reveals that, going back in time and considering countless historical events, national security, in terms of armed conflict, is closely and increasingly related to the cybersecurity of nations themselves.

These observations support the conclusion that cybersecurity appears to be a phenomenon and process that needs to be monitored collectively, as demonstrated by the most recent actions promoted by nations around the world.

1.2 How To Face The Current Cyberthreats?

The cyberwarfare-oriented landscape certainly had its influence in prompting nations worldwide to proactively respond to this incessant conflict. Although the previous section analysed recent military conflicts with significant cyber implications, the same considerations can be made for less recent crises that also involved cyber components.

Nations have started implementing collective mechanisms to protect their security by strengthening the security of individual systems within their territory. This shift is due to the increasing awareness that the vulnerability of infrastructure and systems within national borders can serve as a vector of attack for large-scale attacks.

In particular, a recently adopted *modus operandi* has been the development of *legislative instruments* aimed at regulating the security and privacy of the digital defences of citizens, organisations and critical infrastructures. This paradigm shift meant that security tasks in companies involve not only finding optimal and secure solutions in systems and software but also interfacing with the regulatory environment.

From this strategic evolution, the phenomenon of *compliance* has emerged. As a result, both public and private organisations must be fully aware of the legislation in force, to adhere to the complex and ever-evolving realm of security measures, and thus address the laborious task of compliance verification [87].

Whereas in the past such security measures were strictly linked to internal and organisational processes [11], [123], [64], over the years, the security measures have been designed to also cover technical aspects. However, although the issue of security is common throughout the world, from a regulatory point of view, the problem is treated differently. On the US side, NIST has proposed the Cybersecurity Framework [99], where a series of controls are proposed in a structured manner, and which we could consider as closer to the technical aspects of security.

For example, management of hardware, software, services and security architectures, consistently with the organisation's risk strategy to protect their confidentiality, integrity, and availability; monitoring of networks and network services for finding adverse events. Instead, Europe has shown itself to be at the forefront of cyber and non-cyber risk management, thanks to the consistent promulgation of several types of legal acts, namely Regulations, Directives, Decisions, Recommendations and Opinions [33] of which specific instances are illustrated in Figure 1.4.

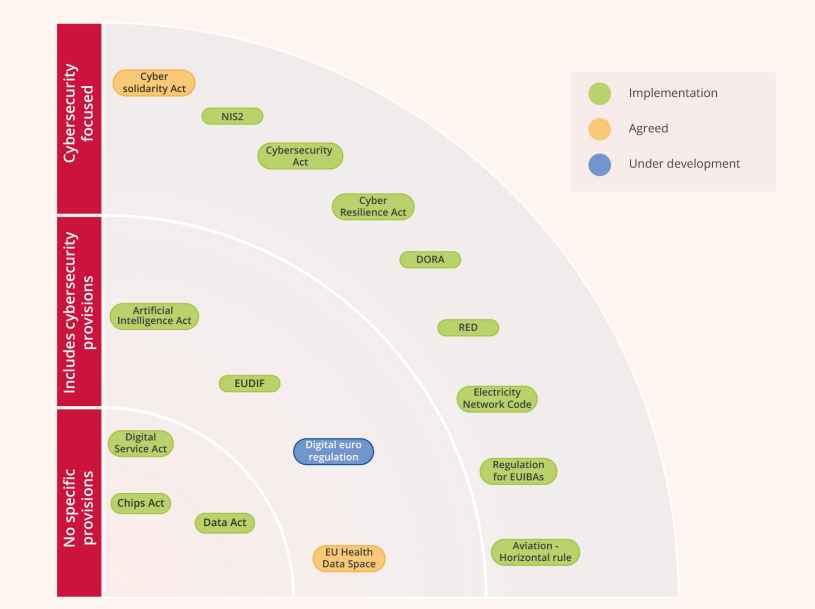


Figure 1.4: ENISA 2024 Report on the State of Cybersecurity in the Union [44]

A noteworthy counterpart to the NIST Cybersecurity Framework is the NIS 2 Directive (Directive (EU) 2022/2555) [46], which establishes a comprehensive set of cybersecurity requirements applicable to both essential and important entities across a wide range of sectors, as well as to Member States. Compared to its predecessor, the NIS 2 Directive significantly broadens its scope, encompassing sectors classified as essential (e.g., energy, transport, healthcare, digital infrastructure, public administration, and space) and important (e.g., postal services, digital providers, electronics, among others).

Unlike the more structured and technically oriented NIST Framework, the NIS 2 Directive adopts a higher-level approach, with less emphasis on concrete technical measures. Its legal language, aiming for comprehensiveness, often results in a complex and verbose formulation that

typically requires interpretation by specialised professionals.

The consequences of non-compliance, which might amount to several million euros, highlight how important and urgent its execution is. However, it is still difficult to translate the general rules of the Directive into detailed, context-specific operational procedures, which frequently necessitate significant financial and human resource investments.

1.3 How Is The Adoption Of Security Legislation Being Responded To?

Organisations, companies, and other entities positively perceive the integration of regulatory security policies. As highlighted in a report presented at the World Economic Forum [140], there is a clear upward trend between 2022 and 2025 regarding the perception that security policies are effective in reducing cyber risks. Some statistics presented in such a report, and illustrated in Figure 1.5, show that although a slight decrease in 2024 of 12 percentage points compared to 2023 can be observed, the leap from 2022 with 39% to 2025 with 78% shows a 39% increase, thus a 100% increase compared to 2022.

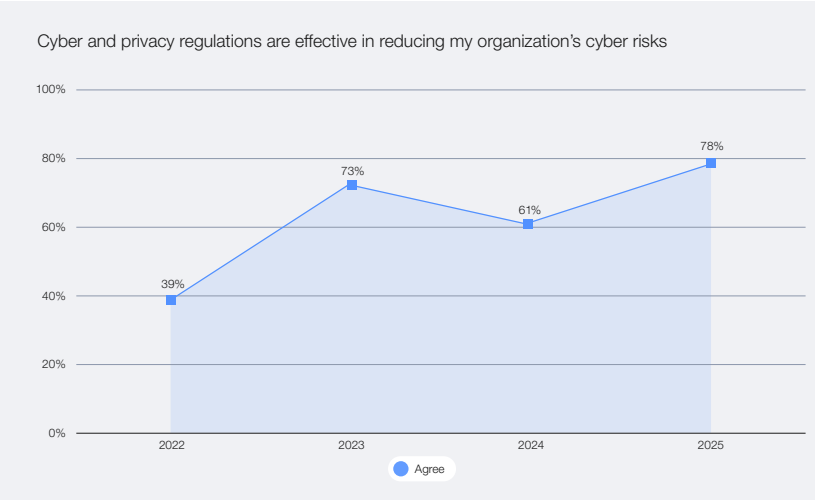


Figure 1.5: World Economic Forum - Global Cybersecurity Outlook 2025 [140]

While regulatory frameworks like the NIS 2 Directive undoubtedly play a role in improving the security status of companies and organisations, they create a conceptual gap with real systems, as they contain

security measures expressed in formal and specific language, which are difficult to interpret even for security experts. Recent reports by KPMG [73], ECSO (European Cybersecurity Organisation) [40] (reported in Figure 1.6) and KnowBe4 [72] (reported in Figure 1.7) acknowledge and highlight these strong limitations. Although it can be assumed that this generality is due to avoiding strict correlation with specific services, products and practices, on the contrary, too much generality provokes technological and scientific hesitation.

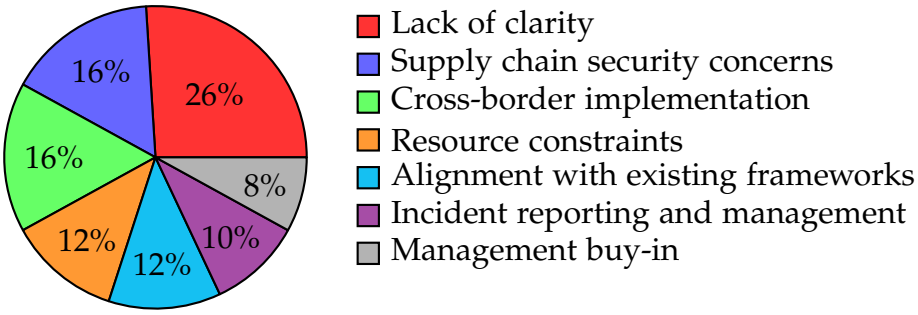


Figure 1.6: Main challenges in the NIS 2 implementation from an organisational perspective [40]

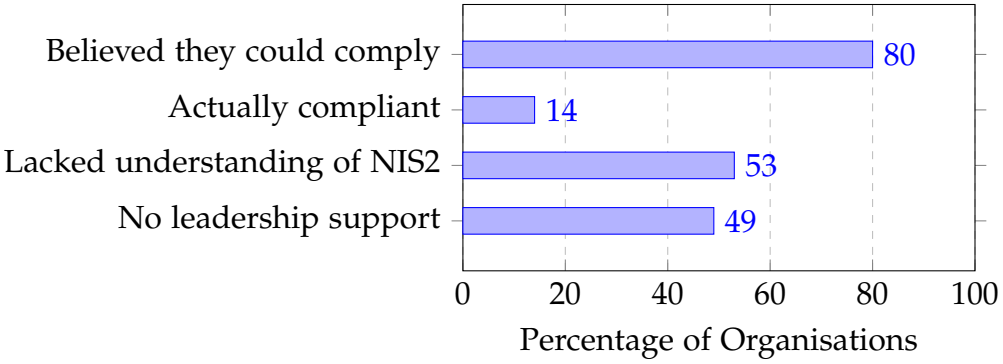


Figure 1.7: Business State of NIS 2 [72]

It can be undoubtedly noted that the *lack of clarity* category — and equally *lacked understanding of NIS 2* — are present with significant percentages. In the first case (ECSO report), the lack of clarity of the Directive appears to be 26% of possible challenges for companies, resulting in being the main challenge. In the second case (KnowBe4 report), on the

other hand, 53% of the respondents believe that the directive lacks clarity.

Only recently (June 2025), ENISA published guidelines for implementing certain NIS articles [42]. However, the following temporal and spatial drawbacks can be identified:

- **Temporal drawback**. Compared to the publication of NIS 2 (end of 2022), these guidelines were only recently published (summer 2025). There is therefore a substantial time gap.
- **Spatial drawback**. As pointed out in the documentation, the guidelines do not replace the measures of the Directive or require their implementation. Therefore, the arbitrariness in their use places them at a lower level of importance than the Directive itself, whose adoption is mandatory.

As the NIS 2 Directive is rather recent, it is not yet possible to collect comprehensive information about possible fines imposed and what the attitude of companies in Europe has been. The statistics already presented, in any case, appear to be preliminary considerations. However, taking advantage of the GDPR age, being a regulation that has already been in place (and consequently adopted) for about ten years in the European landscape, the tables 1.1 and 1.2 present numerical information on the types of violations incurred and the amount of economic penalties resulting. These insights offer a valuable anticipation of how the application of NIS 2 might evolve.

Table 1.1: Summary of GDPR Fines by Violation Type [54]

Violation Type	Total Fine (€)	Fines Count
Insufficient legal basis for data processing	2.507.757.757	719
Non-compliance with general data processing principles	2.506.144.350	684
Insufficient technical and organisational measures to ensure information security	853.893.412	463
Insufficient fulfilment of information obligations	252.723.560	201
Insufficient fulfilment of data subjects rights	102.223.046	235
Unknown	22.848.041	18
Insufficient cooperation with supervisory authority	6.869.569	145
Insufficient fulfilment of data breach notification obligations	3.984.292	47
Insufficient data processing agreement	1.123.110	13
Insufficient involvement of data protection officer	978.000	25
Lack of appointment of data protection officer	25.000	3

Table 1.2: Top GDPR Fines by Controller [53]

#	Controller	Fine (€)	Type of Violation	Date
1	Meta Platforms Ireland Limited	1.200.000.000	Insufficient legal basis for data processing	2023-05-12
2	Amazon Europe Core S.à.r.l.	746.000.000	Non-compliance with general data processing principles	2021-07-16
3	TikTok Technology Limited	530.000.000	Insufficient legal basis for data processing	2025-05-02
4	Meta Platforms, Inc.	405.000.000	Non-compliance with general data processing principles	2022-09-05
5	Meta Platforms Ireland Limited	390.000.000	Non-compliance with general data processing principles	2023-01-04
6	TikTok Limited	345.000.000	Non-compliance with general data processing principles	2023-09-01

It can be observed that *non-compliance [...]* and *insufficient technical and [...] to ensure information security* categories, respectively, are the second and third types by number of violations (Table 1.1). By consequence, among the largest companies in Europe, the category *non-compliance [...]* results in being the most common type of violation (Table 1.2).

These considerations reinforce two key points: the lack of clarity, as well as the historical evidence of non-compliance, are two significant challenges when approaching the regulatory framework.

Building on the presented observations, this dissertation offers a scientific and technical approach to bridge the gap between the regulatory landscape and the more technical aspects of cybersecurity.

1.4 A Dissertation To Explore New Security Horizons

It has been described how the contemporary geopolitical conflicts inevitably affect the digital sphere, the so-called *infosphere*. Nations are subject to increasingly pressing and targeted attacks, forcing them to adopt different strategies aimed at collectively strengthening their cybersecurity posture. One prominent response has been the promulgation of directives, regulations, and generally normative instruments to regulate

collective security activities. Consequently, the concept of compliance takes on greater importance.

From an organisational perspective, compliance is the condition of adhering to the security measures defined in the legislation. Although this regulatory landscape has been positively perceived, several interpretative and implementation challenges have emerged. We can argue that *compliance is not a static state*. Even after the correct implementation of a measure, a switch from compliance to non-compliance status could occur, e.g., due to technological obsolescence [110] or unforeseen vulnerabilities, and new security gaps could emerge. Attackers could be aware of these legislative/implementation gaps and attempt ad hoc *offensive security activities*, as a range of security strategies to test security posture [66]. This leads to the emergence of a refined attacker model — an evolution of the General Attacker model [13] — who exploits legislative shortcomings.

This thesis aims to innovatively bridge the conceptual and technological gap that, somewhat paradoxically, has been created in an attempt to respond to the posed and concrete security problems. The dissertation proposes a scientific path to *deconstruct* high-level security legislation, hence identifying structural and conceptual weaknesses, and developing specific methodologies aimed at strengthening the adoption of the legislation itself.

In particular, this thesis illustrates different methodologies that start with the simple textual examination of security legislation, especially the NIS 2 Directive, and move on to the creation of compliance-checking systems, research of attacks, their organisation into killchains, to the point of even propose strategies for the discovery and confirmation of vulnerabilities as a further step in future research.

Upon considering the made observation and the identified research directions, this dissertation poses the following Research Questions (**RQ**):

- **RQ1 - Initial RQ** *What is the current cyberthreat landscape and how is it being responded to?*

  We answer **RQ1** in the current Chapter (**Chapter 1**), by analysing the current cyberthreat scenario, which sees the spread of attacks, in most cases linked to geopolitical conflicts, primarily targeting entire nations. This motivation, and in general the increasing importance of security, has led nations to propose collective strategies for strengthening their security posture, including the promulgation of common security measures outlined in legislative texts, particularly in ***security directives***. Nevertheless, security directives express security concepts perceived as too general, poor in clarity, leading to a conceptual and technological gap with the technical aspects of security.

- **RQ2** *Can the security measures within security legislation be interpreted for improved interoperability and automated compliance verification?*

  We answer **RQ2** in **Chapter 3** by proposing the `SecOnto` methodology that involves a comprehensive ontological representation of security measures and the corresponding responsible agents. `SecOnto` provides a series of sequential steps for deconstructing the security measures from their simple textual format into an ontological format, to support the automation of the compliance process through the potential of ontological reasoning and the structured research of attacks. `SecOnto` is validated in the NIS 2 Directive.

- **RQ3** *Can attacks be derived from non-compliance with the high-level security measures outlined in security legislation?*

  We answer **RQ3** in **Chapter 5** by proposing WISARD, a methodology for searching for attacks from measures defined in security directives and expressed in legal language. WISARD aims to compensate for this generality by using two *semantic methods* to search a specific category of attacks, namely *attack patterns*: the first method is based on the concept of *semantic similarity*; the second method is based on the concept of *ontological semantics*. The two semantic methods designed in WISARD aim to derive attack patterns from security measures that are *semantically* correlated. Furthermore, to strengthen their correctness, the obtained correlations by the semantic methods are cross-referenced with a validation base obtained with the involvement of security experts. WISARD is applied to derive attack patterns defined in the CAPEC framework from the security measures of the NIS 2 Directive.

- **RQ4** *Can offensive activities be conducted from non-compliance with the security measures outlined in security legislation?*

  We answer **RQ4** in **Chapter 6** by proposing MOSKAD, a methodology for structuring attack patterns into offensive killchains. MOSKAD build upon the attack patterns derivable from WISARD to model each step in constructing ad-hoc killchains, leveraging the non-compliance status of a company with the security measures defined in security directives. The attack patterns are composed and structured into killchains whose individual phases are operationalised by the attack patterns themselves. The attack patterns are correlated with the killchains through the techniques from the ATT&CK framework as intermediate nodes. To identify these correlations, different machine learning techniques have been analysed in order to obtain the *best correlational method*, i.e. the method that best captures these correlations, to build the most correct killchain possible. MOSKAD is demonstrated with the security measures of the NIS 2 Directive.

- **RQ5** *Can non-compliance with the security measures outlined in security legislation be used for the detection of vulnerabilities?*

  We answer **RQ5** in **Chapter 8** by providing `Seer`, a methodology for vulnerability detection in software. `Seer` presents anticipations of possible future research that further strain the concept of offensive security. In particular, the methodology leverages the CWEs (hierarchically linked to the attack patterns obtainable from `WISARD` methodology). Through machine learning, the CWEs are employed to predict whether the software can be affected. Subsequently, fuzzing with automatic harness generation is employed to reduce the false positive rate. Hence, the same CWEs can be confirmed, which eventually leads to the detection of possible existing CVEs.

- **RQ6 - Conclusive RQ** *What key insights does this research provide on deconstructing high-level security legislation, and how can they guide future improvements against cyberthreats?*

  We answer **RQ6** in **Chapter 9** by providing conclusive findings and future research directions. We discuss how the proposed methodologies can enhance the overall security posture by reducing the gap between the offensive security practices and security legislation. Possible research directions and case studies will be analysed, for which this dissertation has set the investigative direction.

Strictly related to the Research Questions, we formulate the following Sub-Research Questions (**SRQ**):

- **SRQ1** *Can the security measures contained in a security directive and the characterising elements defined therein be extracted automatically?*

  We answer **SRQ1** (linked to **RQ2**) in **Chapter 2** by illustrating `GTCheck`, a methodology on the application of Natural Language Processing (NLP) in correctly extracting the grammatical entities of the security measures from security directives. This study faces the challenge of the essential interpretation of the legal language of cybersecurity, namely of the extraction of the essential Parts of Speech (POS). The methodology leverages state-of-the-art open-source NLP tools, as well as manual analysis, to validate the outcomes of the tools. The methodology particularly focuses on the NIS 2 Directive.

- **SRQ2** *Can the NIS 2 Directive be provided with a structured and interoperable design for the enhancement of compliance verification and the specific research of attacks?*

  We answer **SRQ2** (linked to **RQ2**) in **Chapter 4** by presenting `NIS2Onto`, a full ontology of the NIS 2 Directive. By using the `SecOnto` steps and thus the definitive and proposed structures therein, the NIS 2 Directive is fully represented in ontological format in `NIS2Onto`. Through the semantic representation of the NIS 2 entities, relationships, and security measures, `NIS2Onto` enables automated compliance verification, streamlined risk assessments, and effective policy implementation. Within the ontology, a metrical and qualitative analysis is provided through a real case study to witness the robustness and practical applicability of `NIS2Onto`.

- **SRQ3** *How the semantic similarity step presented in* `WISARD` *behave with non security-directives sources?*

  We answer **SRQ3** (linked to **RQ3**) in **Chapter 7** by discussing `Setàd`, as a methodology that rethinks the semantic similarity step introduced in `WISARD` to derive attack tactics from attack techniques within the MITRE ATT&CK framework. This setting enables a direct mapping between techniques and their associated tactics, providing a structured ground truth for statistical evaluation using the main metrics adopted in machine learning. Within `Setàd`, to ensure precision and verifiability, we formalise the algorithms in Isabelle/HOL, reinforcing their specification.

1.5 Developed Methodologies And Key Exploitable Results

The rational flow that relates the main components discussed within this dissertation is illustrated in Figure 1.8.

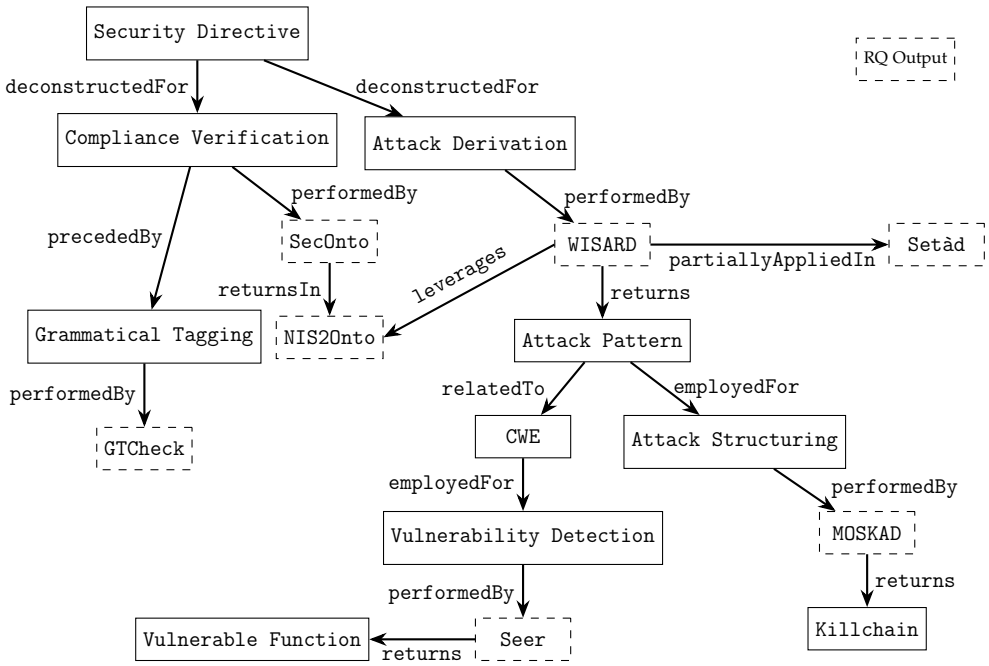


Figure 1.8: Conceptual representation of the methodologies presented in the thesis, and their correlations

Analysis The main components are:

- **Starting Node:** represents the initial information required by the dissertation, such as the *Security Directive*, which serves as the starting point for the entire discussion.
- **Tasks:** indicate operations of transformation, analysis, and/or correlation, applied to the *Security Directive*. The tasks are generalised by the terms *Compliance Verification* and *Attack Derivation*, and also include *Grammatical Tagging*, *Attack Structuring*, and *Vulnerability Detection*. These tasks process inputs or intermediate results, hence outputs.
- **Methodologies:** perform the tasks. The methodologies, as described, are: `GTCheck`, `SecOnto`, `WISARD`, `MOSKAD`, `Setàd`, and `Seer`.
- **Inputs and Outputs:** represent the main outcomes obtained through the applications of the methodologies, or related entities, which may be reusable for other tasks.

Dashed boxes represent the methodologies and products developed within the main and sub-research questions.

Key Exploitable Results (KERs). The key exploitable results comprise the methodologies answering the main RQs and the products derived from their application.

- **KER 1 – Structured Representation**
  - **KER 1A – Methodology:** `SecOnto`, designed to represent, structure, and reason about security directives.
  - **KER 1B – Product:** `NIS2Onto`, the ontological representation of the NIS 2 Directive [56].
- **KER 2 – Attack Correlation**
  - **KER 2A – Methodology:** `WISARD`, designed to correlate attacks leveraging non-compliance with security directives.
  - **KER 2B – Product:** A knowledge base of attack patterns correlated from the security measures of the NIS 2 Directive [22].

- **KER 3 – Killchain Structuring**
  - **KER 3A – Methodology:** `MOSKAD`, designed to structure attacks for offensive activities targeting non-compliance with security directives.
  - **KER 3B – Product:** A software for structuring attack patterns derived from the measures of the NIS 2 Directive into killchains [23].
- **KER 4 – Vulnerability Detection**
  - **KER 4A – Methodology:** `Seer`, designed to detect vulnerabilities considering the non-compliance with security directives.
  - **KER 4B – Product:** A prototype and case study for predicting and confirming vulnerabilities, starting from CWEs.

1.6 Scientific Publications

The content of the methodologies and analyses presented in this thesis has been published in (but is not limited to) the following publications.

Journal Articles

J2 Gianpietro Castiglione, Daniele Francesco Santamaria, Giampaolo Bella, Laura Brisindi, Gaetano Puccia, "Guiding Cybersecurity Compliance: An Ontology for the NIS 2 Directive", In: Computers & Security 157 (2025), p. 104617, ISSN: 0167-4048, https://doi.org/10.1016/j.cose.2025.104617 [29].

J1 Gianpietro Castiglione, Giampaolo Bella, and Daniele Francesco Santamaria. "SecOnto: Ontological Representation of Security Directives". In: Computers & Security 148 (2025), p. 104150. ISSN: 0167-4048. https://doi.org/10.1016/j.cose.2024.104150 [27].

Conference Proceedings

C6 Gianpietro Castiglione and Giampaolo Bella (2025). "Modelling Offensive Security Killchains from Compliance Gaps with Security Directives". In SecAssure - ESORICS 2025 [25].

C5 Gianpietro Castiglione, Marcello Maugeri, Giampaolo Bella (2025). "Poster: Machine Learning for Vulnerability Detection as Target Oracle in Automated Fuzz Driver Generation". In: Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2025. Lecture Notes in Computer Science, vol 15747. Springer, Cham. https://doi.org/10.1007/978-3-031-97620-9_8 [28].

C4 Gianpietro Castiglione and Giampaolo Bella (2025). Compliance-Driven CWE Assessment by Semantic Similarity. In: Garcia-Alfaro, J., et al. Computer Security. ESORICS 2024 International Workshops. ESORICS 2024. Lecture Notes in Computer Science, vol 15264. Springer, Cham. https://doi.org/10.1007/978-3-031-82362-6_24 [24].

C3 Giampaolo Bella, Gianpietro Castiglione, and Daniele Francesco Santamaria. "An Automated Method for the Ontological Representation of Security Directives". In: vol. 3637. 2023. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85185199562&partnerID=40&md5=d744c6a3d320cd56b2192f9b01c0f176 [17].

C2 Giampaolo Bella, Gianpietro Castiglione, and Daniele Francesco Santamaria. "An Ontological Approach to Compliance Verification of the NIS 2 Directive". In: vol. 3637. 2023. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85185195459&partnerID=40&md5=d1896138bfdf5cff023c0b0d45950b83 [18].

C1 Gianpietro Castiglione, Giampaolo Bella, and Daniele Francesco Santamaria. "Towards Grammatical Tagging for the Legal Language of Cybersecurity". In: Proceedings of the 18th International Conference on Availability, Reliability and Security. ARES '23. Benevento, Italy: Association for Computing Machinery, 2023. ISBN: 9798400707728. https://dl.acm.org/doi/10.1145/3600160.3605069 [26].

Table 1.3 illustrates the publication(s) linked to each chapter.

Table 1.3: Chapters with their respective publications

Chapter(s)	3 & 4	3	6	8	5	3 & 4	3 & 4	2
Publication Index	J2	J1	C6	C5	C4	C3	C2	C1

Chapter 2

GTCheck: Grammatical Tagging of Security Directives

The difference between the right word and the almost right word is the difference between lightning and a lightning bug.

— Mark Twain

This chapter is based on the publication "*Towards Grammatical Tagging for the Legal Language of Cybersecurity*" [26].

This chapter presents GTCheck, a methodology for the grammatical tagging of the legislative language. The grammatical tagging leverages state-of-the-art open-source tools for Natural Language Processing (NLP) as well as manual analysis to validate the outcomes of the tools. It is applied to the NIS 2 Directive, providing the first, albeit essential, structured interpretation of such a significant document. This represents the first step in analysing the directive, in terms of the effectiveness of automatically extracting its content for the subsequent methodologies that will be presented.

2.1 Introduction

Legal language, the language from legislative sources, refers to the language typically used in the legal profession, and it may come in both spoken and written form. Recent legislation on cybersecurity often employs legal language in its writing, thereby inheriting all its interpretative complications due to the typical abundance of cases and sub-cases, high-level language, as well as the general richness in detail, although not necessarily sufficient for security operations.

The challenge faced in this chapter is the essential interpretation of the legal language of cybersecurity, specifically the extraction of the main Parts of Speech (POS) from legal documents concerning cybersecurity, such as security directives.

To accomplish this analysis, we propose GTCheck (Grammatical Tagging Check), a straightforward methodology for POS tagging of legal language.

GTCheck seeks to extract at least the relevant clauses from each sentence as well as to identify the relevant POS from each clause, namely the subject (agent), the principal verb or predicate (action), and the object of the action (object). We analysed the 46 articles of the NIS 2 Directive and selected those where actual prescripts are made, leaving aside those about general prerequisites and terminology. We grammatically analysed articles 7 through 37 using both a parser based on ClausIE and manual analysis, concluding that manual validation cannot be avoided over legal language. The analysis indicates that tools such as SpaCy and ClausIE, considered state-of-the-art in NLP-based extraction, reach their limits over the legal language of the NIS 2 Directive, as confirmed by our manual validation of their outputs. Precisely, we found that the tools are good in tagging subjects on average on 96.2% of cases, then verbs on 83.9% and objects only on 64% of cases. Hence, GTCheck requires the analyst to validate their outputs by a systematic comparison with a manual analysis of sentences.

2.2 Related Work

Numerous projects pertaining to law have made use of NLP and machine learning approaches. The expanded capabilities of artificial intelligence have even enabled the employment of other tools to simplify legal context, anticipate verdicts, and search for differences between various legal procedures. This section describes just the most recent and those most closely related to the contributions of this chapter.

The LEGAL-BERT tool adapts the BERT models to the legal context; the main feature is to predict some deliberately maskable words related to the legal context [30]. The scope of the prediction is even broader; Medvedeva et al. investigated the use of NLP for predicting legal decisions, and in particular for predicting if there were any violations of the European Convention on Human Rights [89]. At the same time, Katz et al. constructed a model, based on the random forest approach, for predicting, using only available data, years and years of decisions by the Supreme Court of the United States [70].

Particular attention has been given to the use of NLP in the context of patents: Arts et al. use “natural language processing techniques to harness the rich content of patent documents, identify new technologies

and their impact... validation studies support the use of text mining techniques to identify new technologies and measure patent novelty at the time of filing, and to measure the impact of these new technologies on subsequent innovation." [14]; while Sheng Lee et al. developed a deep learning pre-trained model for the generation of patent claims [78].

NLP techniques have not only been used for prediction purposes and text generation but also for questioning and answering. Zhong et al. presented a question-answering dataset for the legal domain that collects questions from the National Judicial Examination of China [146]. They analysed the dataset to address the challenges of word matching, concept understanding, and multi-paragraph reading. Similar classifications for legal documents were built by de Araujo et al. with the additional feature of leveraging NLP for theme assignment and labelling, regarding Brazilian Supreme Court [86], and by Sulea et al., which apply machine learning techniques for predicting and investigating the sentences of the French Supreme Court [128].

Artificial intelligence techniques have also been used to solve specific instances of inconsistency. Xu et al. developed a framework for legal judgment prediction. tasks [141]. The goal of the framework is to face confusing charges by analysing similar articles and fact descriptions, thanks to a new graph distillation operator. Ul Hassan et al. leverage NLP and machine learning algorithms for legal text classification [134]. They introduced new models for identifying and extracting, from construction contracts, the two categories of requirements and non-requirements. Lippi et al. examine the terms of service of online platforms. In particular, they leveraged machine learning for detecting if any unfair clauses for the user of the online platform exist [82].

In the mere context of information extraction, two main branches can be identified: POS tagging and topic modelling. In the first case, several libraries for different programming languages exist, for example, Scikit-learn [107]; in the second case, Latent Dirichlet Allocation (LDA) is one of the most used approaches [67]. LDA, introduced by Blei, Ng and Jordan [19], is a generative probabilistic model and represents documents as random mixtures over several topics, where a distribution among different words produces a topic.

From the above assessment, it would appear reasonable to conclude that the primary works are predicated on the ad hoc construction of machine learning or natural language processing algorithms inside a particular use case. Specifically, the key objectives of the main strategies are categorisation of text and situation prediction. Although these methods are unquestionably legitimate, they have the drawback of being mostly

reliant on pre-trained models relevant to the particular use case, which necessitates the availability of enormous volumes of labelled data and, as a result, high-performance hardware.

2.3 Special Focus On SpaCy

There have been no notable attempts to apply POS tagging to legal language specifically, and it is clear from the just summarised related work that POS tagging is only a subset of NLP. This chapter aims to create techniques that are lightweight and general enough to be used with nearly any legal language in light of these discoveries.

In the context of the work presented in this chapter, we shall use what is perhaps the best-established tool for processing text, namely the SpaCy library [65], whose main peculiarity beyond POS is Named Entity Recognition.

SpaCy is a free Python toolkit that is one of the most adaptable tools for bringing natural language processing (NLP) to practical applications. We can differentiate between subjects, verbs, and objects to get the right token thanks to SpaCy's syntactic dependency labels. SpaCy is context-independent, working with both local data and pre-trained pipelines linked to a certain language. The SpaCy pipeline begins with tokenisation of the input words. The pre-trained models are then used to tag each token to a specific speech segment, predicting the closest role of each token retrieved from the input.

In practice, it is helpful to use the ClausIE (claucy) citeclaucy sub-library of SpaCy, which can separate the input text into clauses, which are the primary parts of each statement. As we will see in a few examples below, even the most potent, cutting-edge NLP technologies may find it difficult to handle the intricacy of legal language. Furthermore, we are aware that automated information extraction from such a language is a process that begins with POS tagging. Its semantic interpretation is anticipated to be considerably more difficult at the same time.

2.4 The GTCheck Methodology

As we will see below, the process for grammatical tagging any document produced in legal language uses a waterfall structure and is finished with hand validation. To put it briefly, given a legal document, it receives some text written in a legal language as input and promptly analyses

it to produce the POS tagging for each section of the text. Figure 2.1 illustrates the steps involved.

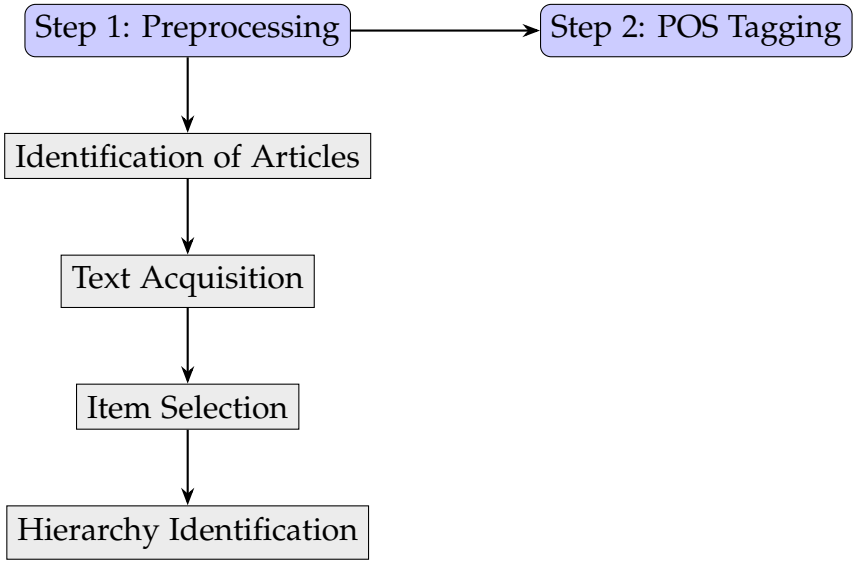


Figure 2.1: Preprocessing and POS tagging workflow

Step 1. Preprocessing To automate document parsing, a preliminary preprocessing phase is necessary. The target is not just information extraction but includes automating the selection of each tagged sentence. Since each legal document has its structure, the parsing must necessarily be manually preset and context-dependent. In general, we want to accomplish the following steps:

a) Identification of articles Assuming we are analysing the legal language of a document that is composed of articles, this step consists of identifying the more significant ones where we can extract relevant POS. These are usually those articles where the actions to be carried out by the entities defined in the document itself are described in precise detail. By contrast, the target articles do not include those with abbreviations and general provisos. It follows that the correct articles can only be determined with a manual and subjective selection of the range to be considered. If we parse all the articles without distinction, we could get an inconsistent collection of the POS, because some might not represent the notions of actions we aim to distil.

b) Text acquisition After finding the range of articles to examine, this step consists of acquiring all the text from a given Article with the aim of automated processing. We shall expect a subdivision into items, which can be numeric or alphabetical, within each article, a style that is quite common in legal documents. A useful acquisition heuristic is to select all text between the string "Article *i*" and "Article *i+1*" to collect all text of the *i*-th article, of course, having checked that such keywords do not occur in the text forming the body of the article. Moreover, if the selection involves two different pages, then spurious information must be deleted.

c) Item selection To input a text, such as a legal one, to any NLP technique, it is necessary to define a heuristic for the choice of the *single units of processing* inside the given text, namely the fragments of text that, given as a whole to the chosen technique, leads to producing new, relevant information. By contrast, applying it to a whole article, which may consist of several sentences, may not yield valuable insights. As the aim is to identify single subjects, actions and objects, the single unit of processing is represented by each single sentence.

d) Hierarchy identification Sometimes, the single unit of processing may have to be built by copying parts of the statement, for example, the subject and action, which may have been omitted to limit redundancy. This is typically the case when a whole itemised sub-list specialises an item from an itemised list. In such a typical case, this step associates the underlying item with the overlying one, namely by removing the colon and linking the underlying hierarchical level with the one above. However, it is necessary to manually verify the unit of processing that is built this way to minimise the risk of inconsistencies.

Step 2. POS tagging As mentioned, grammatical tagging of a sentence extracts the parts of speech and, first of all, identifies the clauses from each unit of processing. Once clauses are identified, each clause may contain up 5 different sentence patterns: Subject (S), Verb (V), Complement (C), Object (C), Adverb (A). In particular, the difference between C and O relies on the role of each of them: a Complement completes the meaning of a sentence while an Object is specifically the direct object on which the action of the verb is reflected. Of course, while only one S and V pair should be expected per clause, several C and O instances may be encountered.

For example, patterns can be extracted by the code snippet Code 2.1, which leverages the chosen libraries. It can be seen that, firstly, it is necessary to load the pre-trained model, and the choice of *en_core_news_lg* means adopting the largest model consisting of 514k keys for a total of 560 MB. In all steps of the method, this represents the highest computational cost, but it is arguably necessary to deal with as many cases as possible. The pre-trained model is then loaded into the pipeline, and the "engine" for text pattern extraction is ready. The code returns the extracted clauses together with all sentence patterns.

Code 2.1: Python code snippet for the extraction of sentence clauses and patterns based on SpaCy and ClausIE

```
import spacy
import claucy

nlp = spacy.load("en_core_news_lg")
claucy.add_to_pipe(nlp)

def NLP(text):
    document = nlp(text)
    return document._.clauses,
        document._.clauses[0].to_propositions(as_text=True)
```

The same thing can be programmed differently, as illustrated in Code 2.2 [101]. The main difference is that Code 2.2 leverages SpaCy only. For the object extraction, the dependency tree generated by the tagging step will be parsed. Only when the tag related to the object is obtained, the sub-text identified by the related sub-tree will be returned. This can be limiting and less precise than using ClausIE, since the latter uses a search based on sentences rather than tags. The preliminary experiments support this claim; it could happen that the objects identified by ClausIE would not be identified by SpaCy alone, indicating that the first approach is preferable to the second, while the performance difference is negligible.

Code 2.2: Python code snippet for the extraction of sentence clauses and patterns based on Spacy only

```
def get_object_phrase(doc):
    for token in doc:
        if ("dobj" in token.dep_):
            stree = list(token.subtree)
            a = stree[0].i
            b = stree[-1].i + 1
            return doc[a:b]
```

Step 3. Tabulation For a correct statistical analysis of the validity of the method, relevant data must be collected and structured. The template that will be filled in for each article is shown in Table 2.2. For each relevant element we extracted by NLP functions in the previous step, we dedicate three columns, for a total of 10 columns, considering an additional one for the identification of the single item. In particular, the first column features the correct POS that we extracted by a manual analysis of the sentence. The second column states the POS extracted using the NLP libraries. At the same time, the third column indicates an evaluation of the correctness of the automated extraction concerning the manual one, using an alphabet of the symbols, represented in Table 2.1. In particular, a tick symbol means that the two analyses coincide. A cross and a vertical bar symbol, respectively, mean that we found the automated analysis to be entirely wrong or partially wrong. Additionally, a "P" symbol stands for clauses in passive style.

Table 2.1: Legend of symbols

Symbol	Acronym
✔	Correct answer
✘	Wrong answer
I	Correct but incomplete answer
P	Passive Verb

Table 2.2: Table template for grammatical tagging of an article

N.	Sub	I-Sub	Sub-HIT	Verb	I-Verb	Verb-HIT	Obj	I-Obj	Obj-HIT
...	...	...	...	...	...	...	...	...	...
	Sub-HIT % = %			**Sub-HIT % = %**			**Obj-HIT % = %**		

The last row of Table 2.2 refers to the percentage of validity of the solutions found by the automated approach. It could be calculated by a simple metric, for example, by assigning full value to the ticks, no value to the crosses, half value to the bars and no value to passive clauses.

2.5 GTCheck Applied To The NIS 2 Directive

This section illustrates the application of the method to the NIS 2 Directive. For the sake of demonstration, the method is illustrated on Article 11 and Article 23 because they cover different cases.

For each step, the following extracts will be analysed:

> 2. Member States shall ensure that their CSIRTs jointly have the technical capabilities necessary to carry out the tasks referred to in paragraph 3. Member States shall ensure that sufficient resources are allocated to their CSIRTs to ensure adequate staffing levels for the purpose of enabling the CSIRTs to develop their technical capabilities.
> ...
> 8. At the request of the CSIRT or the competent authority, the single point of contact shall forward notifications received pursuant to paragraph 1 to the single points of contact of other affected Member States.

Step 1. Preprocesing By coherently applying the various sub-steps, we have the following outcomes:

a) Identification of articles In the NIS 2 Directive, the articles are numbered from 1 to 46. However, not all of them are significant from the point of view of the measures. All articles until 7 are preliminary considerations and definitions, while from the 38 onward various considerations on the applicability of the Directive are described. Therefore, we will parse the Directive from Article 7 to Article 37.

b) Text acquisition To acquire the text of Article *i*, the more efficient solution in the context of the NIS 2 Directive is as anticipated above, namely to grab the text between each pair of article headers of the form "Article *i*" and "Article *i+1*" as strings. Moreover, we must make sure to expand each header with two characters of line feed, both at the start and at the end, because some articles could be cited through some items both by the numerical notation and by their full title. This scenario would hinder the parsing. By contrast, stating which and where the line feed characters are ensures that we capture an occurrence at the beginning of an article, namely, an actual header.

An emblematic case in which the described heuristic is necessary is represented by the following extract:

> Article 10
> **Computer security incident response teams (CSIRTs)**
>
> 1. Each Member State shall designate or establish one or more CSIRTs. The CSIRTs may be designated or established within a competent authority. The CSIRTs shall comply with the requirements set out in Article 11(1), shall cover at least the sectors, subsectors and types of entity referred to in Annexes I and II, and shall be responsible for incident handling following a well-defined process.
> ...
> 10. Member States may request the assistance of ENISA in developing their CSIRTs.
>
> Article 11
> **Requirements, technical capabilities and tasks of CSIRTs**

c) Item selection Coherently with the definition of this step, a single sentence is selected by referring to its termination through the full stop, in the simplest case. The errors that could arise through this approach concern the case in which the full stop does not terminate the sentence. This case only happens in the case of hierarchical lists and is relatively frequent in the NIS 2. It will be treated below.

d) Hierarchy identification In the NIS 2 Directive, each item could have up to two hierarchical levels of depth. The first is listed by letters, while ordinal numbers list the second. Developing a parser that identifies such cases is not complex, since bulleted lists can be used as selection criteria. By contrast, it may be challenging to understand what is represented at the lowest hierarchical level. In the simplest case, we can resolve the following form of compression (Article 9, item 4):

> *That plan shall lay down, in particular: (a) the objectives of national preparedness measures and activities;*

by making the preliminary part of the statement explicit in all possible items, for example, as follows:

> *That plan shall lay down, in particular, the objectives of national preparedness measures and activities;*

In such a case, it generally happens that the primary sentence expresses the presence of some tasks that are described at the first hierarchical level. Consequently, the first hierarchical level states significant verbs that describe the specific actions of the defined subject. Therefore, the second hierarchical level aims at further completing the meaning, defining the objects that the action affects.

Step 2. POS tagging By executing Code 2.1 on some sentences of the previous extracts, the following outputs are produced:

Code 2.3: NLP output on Article 11

```
[<SVC, Member States, shall ensure, None, None, that their CSIRTs
    jointly have the technical capabilities necessary to carry out
    the tasks referred to in paragraph 3, []>, <SVO, their CSIRTs,
    have, None, the technical capabilities necessary to carry out
    the tasks referred to in paragraph 3, None, [jointly]>]
```

Code 2.4: NLP output on Article 23

```
[<SV, the single point of contact, shall forward, None, None,
    None, []>, <SV, the single point of contact shall forward
    notifications, received, None, None, None, [At the request of
    the CSIRT or the competent authority, pursuant to paragraph 1,
    to the single points of contact of other affected Member
    States]>]
```

In the present work, the desired combinations of sentence patterns are SVO or SVC because in these combinations, all the necessary POS are present for a subsequent semantic interpretation. If complex sentences are analysed, the combinations given as output may not be unique and definitive, as in the cases above, where two patterns were extracted for each article. Therefore, it would be necessary to choose the most appropriate pattern manually.

In consequence, obtaining only pattern SV may be taken to signify incompleteness of the automated approach, while pattern SVOC falls into the cases that can be considered complete. The whole output consists of sentence pattern type, sentence subject, sentence verb, sentence indirect object, sentence direct object, sentence complement and sentence adverbials. The experiments indicated that all possible combinations of the 5 can be output on the NIS 2.

It can be seen from Code 2.3 that two solutions were generated, both containing the object, but only the first solution is complete.

The correct Subject was identified, together with the Verb and the Complement. Although the second identified sentence pattern might have seemed reasonable, the POS identified are not accurate.

Code 2.4 shows that the produced sentence pattern SV indicates no object was identified. While we can see that both identified sentence patterns produced the correct Subject, the second sentence pattern is wrong in the handling of the verb. This further confirms that the automated approach is not perfect.

A significant case that was encountered is the presence of different hierarchical levels with a specific feature: the object of the sentence introducing the hierarchical level explicitly expresses their presence. This leads to inconsistencies. For example: *Member State shall have the following tasks: a) . . . b) . . . .* There is greater complexity towards grammatical tagging in the presence of the second hierarchical level.

Step 3. Tabulation Iterating the extraction over all relevant Articles allows us to collect all the information we aim at. The POS of articles 11 and 23 are depicted respectively in Table 2.4 and Table 2.5, while the used acronyms are defined in Table 2.3. The symbols purposely differentiate the incomplete answer from the incorrect answer, since the first one is to be considered partially correct. Therefore, its extraction does not have to be considered entirely misleading.

Table 2.3: Legend of acronyms

Name	Acronym
Member State	MS
National Cybersecurity Strategy	NCS
Competent Authority	CA
Commission	Co
Point of Contact	POC
CSIRT	C
ENISA	E
Cooperation Group	CG
The European External Action Service	EEAS
CN	CSIRTs Network
Eu-Cyclone	EuC
SYS	Supporting Information Systems

Table 2.4: Table for grammatical tagging of NIS 2 Article 11

N.	Sub	I-Sub	HIT	Verb	I-Verb	HIT	Obj	I-Obj	HIT
1	-	-	-	-	-	-	-	-	-
1.a	C	C	✔	ensure	ensure	✔	**N11.1.2.a**	**S11.1.2.a**	❙
1.b	C Premises and SYS	C premises	❙	P - located	located	✔	**N11.1.2.b**	**S11.1.2.b**	✘
1.c	C	C	✔	P - equipped	equipped	✔	**N11.1.2.c**	**S11.1.2.c**	✘
1.d	C	C	✔	ensure	ensure	✔	**N11.1.2.d**	**S11.1.2.d**	✔
1.e	C	C	✔	P - staffed	ensure | ensure	✔	**N11.1.2.e**	**S11.1.2.e**	✘
1.f	C	C	✔	P - equipped	equipped	✔	**N11.1.2.f**	**S11.1.2.f**	✘
2.1	MS	MS	✔	ensure	ensure	✔	**N11.2.1**	**S11.2.1**	✔
2.2	MS	MS	✔	ensure	ensure	✔	**N11.2.2**	**S11.2.2**	✔
...	...	...	...	...	...	...	...	...	...
	Sub-HIT % = 95 %			**Sub-HIT % = 90 %**			**Obj-HIT % = 45 %**		

Table 2.5: Table for grammatical tagging of NIS 2 Article 23

N.	Sub	I-Sub	HIT	Verb	I-Verb	HIT	Obj	I-Obj	HIT
1.1	MS	MS	✔	ensure	ensure	✔	**N23.1.1**	**S23.1.1**	✔
1.2	Entities concerned	Entities Concerned	✔	ensure	ensure	✔	**N23.1.2**	**S23.1.2**	✔
1.3	MS	MS	✔	ensure	ensure	✔	**N23.1.3**	**S23.1.3**	✔
1.4	Mere ... notification	Mere ... notification	✔	not subject	subject	✘	**N23.1.4**	**S23.1.4**	✔
...	...	...	...	...	...	...	...	...	...
8	POC	POC	✔	forward	NONE	✘	**N23.8.1**	**S23.8.1**	✘
9.1	POC	POC	✔	submit	submit	✔	**N23.9.1**	**S23.9.1**	✔
9.2	E	E	✔	contribute	contribute	✔	**N23.9.2**	**S23.9.2**	✔
9.3	E	E	✔	inform	inform	✔	**N23.9.3**	**S23.9.3**	✔
10	C	C	✔	provide	provide	✔	**N23.10.1**	**S23.10.1**	✘
11	Commission	Commission	✔	adopt	adopt	✔	**N23.11.1**	**S23.11.1**	✘
	Sub-HIT % = 92.8 %			**Sub-HIT % = 80.9 %**			**Obj-HIT % = 83.3 %**		

The tables exhaustively list the occurrences of relevant extracted POS. The tables provide a clear and first-level view of the NLP results previously discussed. Naturally, more correct results are obtained where there is less complexity of the related POS; therefore, in most cases, simplicity concerns the subjects.

2.6 Evaluation Of `GTCheck`

To evaluate the overall functioning of the automated tagging, hence `GTCheck`, compared to the manual one, we average the hit values for each

tag of each grammatical tagging table. As suggested above, we associate hit value 1 with a correct answer, 0.5 with an incomplete answer and 0 otherwise. Therefore, for each table i and for each tag j, the Hit Rate (HR) is the sum of the HIT values divided by the number of occurrences:

$$HR_{i,j} = \frac{\sum HV_{i,j}}{\#occ_{i,j}} \cdot 100, \ \textit{where } 7 \leq i \leq 37, 1 \leq j \leq 3.$$

We decided to skip the following cases: measures with no item number, measures that may include a priori contextualisation and hierarchical levels introduced by "have following tasks" and similar ones where no other agent is expressed. The reasons are, respectively: difficulty in automatically parsing the measure since it has no identifier (for example, Article 17); information extraction is sometimes wrong because the libraries focus on the contextualisation clause; the extracted information is often incorrect because from the moment the clause is compressed with the higher hierarchical level, inconsistencies are formed (except in the case where the additional level contains its subject that allows generalising to the main case of subject, verb and object). Therefore, in cases that have structural rather than semantic problems, since the results would always be wrong and would have consistently lowered the percentage, we preferred to consider them exceptions. The resulting percentages are shown in Table 2.6, with a graphical representation in Figure 2.2.

As can be seen, the highest values are obtained over the Subject and Verb. It may be argued that Objects tend to have lower HIT rates because they may be more complicated. The cases in which they are rather good, compared to the verb of the same item, on the other hand, are due to the presence of hierarchical levels that involve a greater number of sub-cases whose summation raises the value.

Table 2.6: Hits of the automated approach wrt the manual one

Art.	Sub-HIT	Verb-HIT	Obj-HIT
7	100%	100%	91.6%
8	100%	88.8%	61.1%
9	100%	100%	43.3%
10	100%	93.3%	60%
11	95%	90%	45%
12	100%	57.1%	58.3%
13	85.7%	85.7%	50%
14	90%	100%	50%
15	95%	80%	43.8%
16	100%	87.5%	66.6%
17	–	–	–
18	100%	50%	75%
19	100%	83.3%	57.5%
20	75%	75%	100%
21	90%	60%	60%
22	100%	100%	50%
23	92.8%	80.9%	83.3%
24	80%	80%	40%
25	100%	100%	50%
26	100%	33.3%	33.3%
27	100%	75%	83.3%
28	100%	91.6%	86.6%
29	100%	100%	66.7%
30	100%	100%	100%
31	100%	85.7%	42.9%
32	100%	75%	67.1%
33	93.8%	75%	90%
34	90%	70%	65%
35	100%	100%	50%
36	100%	100%	50%
37	100%	100%	100%
Average	**96.2%**	**83.9%**	**64.0%**

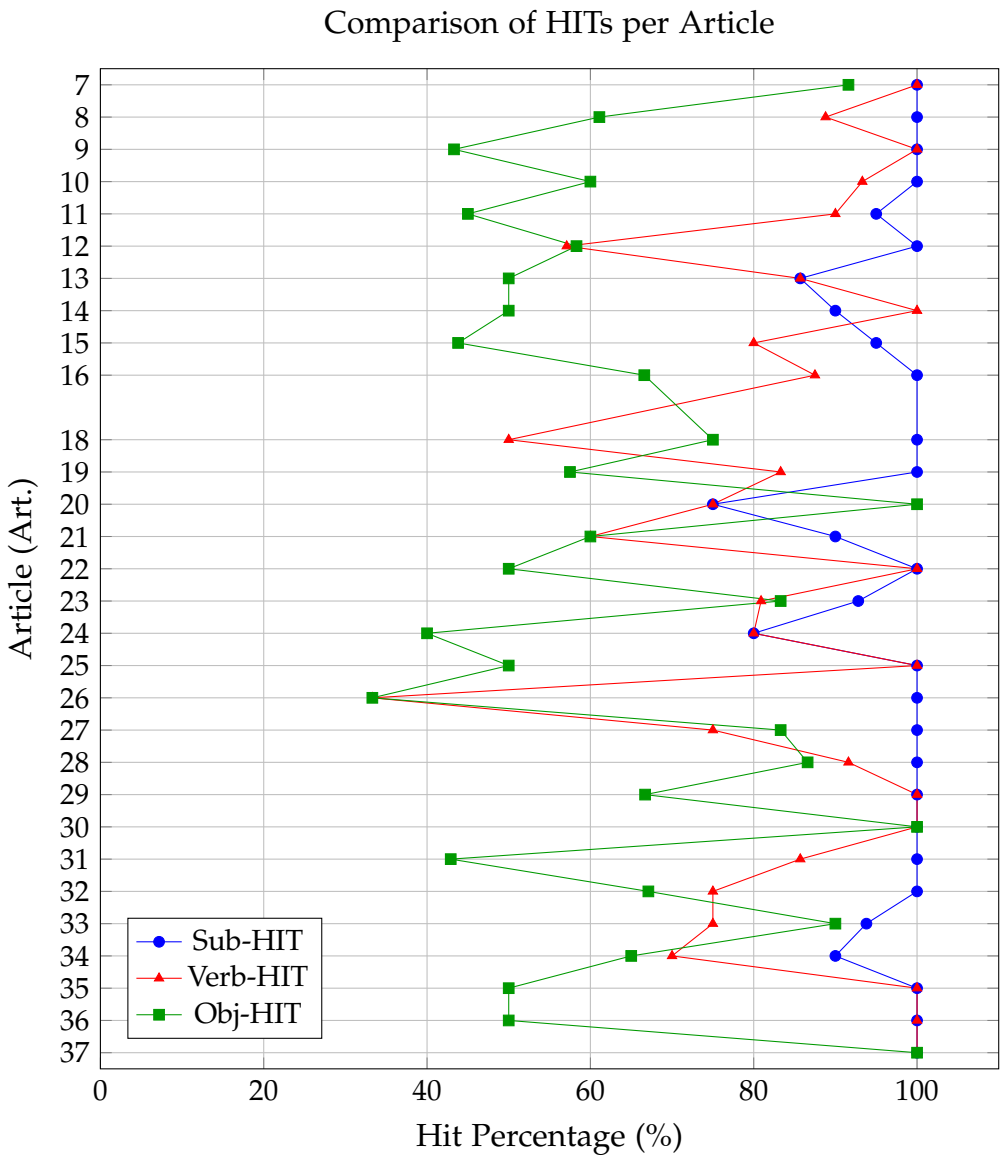


Figure 2.2: Hits of the automated approach with respect to the manual one

2.7 Identified Open Challenges

As demonstrated by the statistical analysis, challenges can be encountered, with a certain relevance, through all cases of information extraction.

Identification of the Object The challenge of this type does not stem from the complexity of a single word: the pre-trained model is very good

at identifying the English words encountered in the text. This can be demonstrated with a single and simple part of speech tagging.

On the contrary, it is difficult for the NLP library to extract the clauses and associate them with the correct place in the sentence. An improvement here would be a breakthrough since one could remove the sentences that simply lengthen the sentence, without adding much additional information, and keep only those sentences that provide significant information.

Associating vast portions of text to either Complement (C) or Object (O) as it currently works limits extraction twice: even irrelevant details are associated, causing an overly general tag, and, at the same time, the specificity of the individual clauses is lost because many clauses are often tagged together.

Identification of the style An article may begin with a lengthy contextualisation preamble. This is not infrequent in legal language and serves to contextualise the specific conditions for the application of the very measure that is described in the sequel of the article. This turns out to be a challenging case for the automated approach. The POS are often extracted from the early parts of the sentence, with the result that they are obtained from the preamble rather than from the actual measure in this case, hence they are incorrect.

Grammar issues Among these, we find the management of passive verbs. In their presence, the object is rarely identified. Managing passive contexts is still a substantial limitation for SpaCy. Another problem is the use of pronouns. Of course, this is not exclusively connected to NLP, but the association of nouns with their respective pronouns indeed remains an open problem, particularly in the context of subjects.

2.8 Automating `GTCheck`

`GTCheck` cannot be considered to be fully automatable since, at its core, it evaluates without the availability of a ground truth for a full automatic evaluation.

In any case, the first part of selecting the security measures must be delegated to human scrutiny, which decides the relevant ones in consideration of their security assessments. As a consequence, the validation of `GTCheck`, hence of the NLP tools it employs, we find should remain a manual task, necessary for evaluating the tools themselves.

This is particularly relevant in our analysis of legislative sources, where the security measures may be very difficult to interpret even for the modern language-processing tools.

2.9 Concluding Remarks

This chapter contributed to the essential interpretation of documents written in legal language by advancing an automated methodology for their grammatical tagging. GTCheck leveraged the state of the art in the area of NLP, and it is arguably applicable to any target legal document following some preprocessing tailored to the specific features of the target.

The application of GTCheck to the NIS 2 leads to the second finding, which consists of the first, albeit essential, structured interpretation of the entire directive. We have built all tables for the grammatical tagging of articles from 7 through to 37, showing the POS produced by leveraging the NLP libraries next to those produced manually by the analysis and, finally, the relevant hit rates [55].

A more technical finding is that the forefront libraries SpaCy and ClausIE are compelling and offer remarkable help. However, they are clearly put at stake by the complexities of the legal style. As a result, while their hit rates are excellent over subjects and good over verbs, they are inadequate over objects. Therefore, automated extraction with NLP must still be supported by human intervention.

Chapter 3

SecOnto: Ontological Representation of Security Directives

The limits of my language mean the limits of my world.

— Ludwig Wittgenstein

This chapter is based on the publication "*SecOnto: Ontological Representation of Security Directives*" [27].

This chapter introduces SecOnto, a methodology composed of five steps for providing a structured and interpretable representation of security directives through ontologies. SecOnto is described and validated through operational examples based on the NIS 2 Directive.

3.1 Introduction

Security directives often comprise complex, interrelated measures that must be meticulously deciphered, understood, and implemented. Complications that, in most cases, as already illustrated in Chapter 2, also lead to errors in even the most modern text analysis tools.

In response to the growing demand for clarity and guidance, the work in this chapter proposes the adoption of Semantic Web ontologies for interpreting and operationalising security directives. Ontologies, as a structured mathematical instrument for representing knowledge, offer an innovative and systematic approach to comprehending the intricate web of security regulations, in particular for the so-called Regulations as Code (RaC) concept; RaC aims at encoding and publishing legislation, regulations, and policies as machine-readable code; hence, it finds breeding ground in the Semantic Web context [85].

Ontologies and their semantic approach to knowledge organisation enable more precise and systematic modelling of complex documents in the legal area, with an emphasis on delineating and formalising the key components contained in the document, in line with the concept of Legal

Informatics. By representing these components as discrete entities with defined relations, ontologies improve the ability to navigate and structure complicated legal and security texts. These kinds of assumptions are appropriate for organisations because they can lead to 1) compliance verification procedures that are more effective, precise, and auditable; 2) targeted and structured attack derivation.

3.2 Related Work

The application of ontologies in the context of compliance management, as stated by Yip et al. [142], and ontology-based modelling, has led to extensive literature and research and underscores the critical role of ontologies and reasoning in improving understanding and compliance in general. It is also worth mentioning that, outside ontologies, several approaches for policy formalisation have been proposed, and some works will be shown.

Schmidt et al. [120] developed an ontology-based approach for checking the compliance of service processes (the authors stated that a service process is a process that produces a service). Cheng et al. [32] presented the enhancement and population of a compliance ontology through the use of information extraction, in order to facilitate semantic interoperability and lessen the complexity and duplication of tasks involved in compliance monitoring for an enterprise. The information is obtained from an existing GRC system (Eramba). In the work of Zhong et al. [145], automated compliance checking and environmental monitoring are deployed within a distributed energy station project.

Many of the efforts have been particularly dedicated to addressing the diverse challenges posed by security and privacy issues, within legal and security documents, especially regarding the General Data Protection Regulation (GDPR), one of the pioneering data protection regulations. Among the many possible works, some stood out.

Bartolini et al. have worked extensively on the GDPR encoding, in particular using Reified Input/Output Logic, for example, to be able to correlate ISO/IEC 27018:2014 with GDPR [16]. In the work of Elluri et al. [41] two legal documents are integrated (GDPR and PCI DSS) through an ontology to highlight the common characteristics of user's data protection guidelines and after leveraging reasoning for the organisation compliance verification of obligations. Palmirani et al. [103] presented PrOnto, a legal ontology on the GDPR. The PrOnto's goal is to provide a "legal knowledge modelling of the privacy agents, data types, types

of processing operations, rights and obligations". Hasan et al. [61] presented the compliant information system development (CISMET) ontology, which instantiates the General Data Protection Regulation (GDPR) to show how regulatory requirements compliance concepts are linked to system development initiatives. Debruyne et al. [34] used ontologies for representing GDPR consent concerning data processing purposes. The datasets are mapped into RDF schemas (an ontological language for describing metadata) for subsequent compliance verification. Bonatti et al. [20] proposed a similar work in the context of the European project SPECIAL, including a policy language for expressing consent policies and obligations, and approaches for verifying if data processing is compliant with the consent provided by subjects. Pandit et al. [104] provided a proof-of-concept example where data is retrieved and automatically populated to fill out the GDPR preparation checklist made public by Ireland's Data Protection Commissioner. Rahmouni et al. [113] proposed a method based on multiple technologies in the semantic Web stack to address conflicting legal and ethical frameworks within a single EU Member State as well as between different national frameworks may apply to privacy needs like patient consent in the context of health grids. Fenz [47] proposed a methodology for producing IT-security metrics based on ISO 27001 automatically, usable by organisations that can assess both the efficacy of control implementations and their compliance with information security requirements.

Drogkaris et al. [39] worked on privacy rules that maintain adherence to the underlying legal and regulatory environment. The work suggests a multitier strategy that leverages current governmental hierarchies to accommodate varying data requirements and service provider-imposed processing requests, while preserving compliance. Oltramari et al. introduced PrivOnto [100] a semantic framework designed to describe privacy regulations using annotations. PrivOnto analysed 115 US-based organisations' privacy policies, deriving a corpus of over 23,000 annotated data practices. The goal of PrivOnto is to address user-relevant privacy inquiries and to assist scholars and authorities in the comprehensive examination of privacy regulations. Breaux et al. [21] devised a methodology to translate a rigorous subset of often encountered privacy needs from natural language text to Eddy, a formal language in description logic. Developers can enable data flow tracing inside policies and identify inconsistencies between privacy standards by using this language. To extract privacy needs from policy documents, the authors developed a set of heuristics.

Instead, in more general terms regarding the role of an agent in the

security measures, several years ago, a formal framework for modelling and analysing security needs in their social and organisational context, called Secure Tropos, was introduced by Giorgini et al. [57]. Several notions are put out by Secure Tropos, including an actor that combines the ideas of a role and an agent, a goal that can be honed by breaking it down into smaller, more specific sub-goals, a job, and a resource.

Last but not least, it is worth mentioning the work proposed by Distinto et al. [37], in the context of the European directives. The paper describes the development of the legal ontology named LOTED2. The work focuses on setting European public procurement notices within their legal framework, also tracking initiatives about the construction of linked data-compliant representations of information surrounding tender notices in Europe. LOTED2 is intended to facilitate the development of Semantic Web applications.

All these works approach specific security measures, and, in general, they do not aim to provide a methodology for security measures described as text to be interpreted ontologically, with specific steps for examining and analysing the style of the measures in detail using a generalizable design. Therefore, they do not seem entirely suitable and scalable for the disambiguous interpretation of the legal language of modern security directives.

The goal of the present chapter is to define an ontology-based methodology that is potentially verticalisable to any legal security document; therefore, it must be replicable, with the benefits that the same method aims to bring.

3.3 Special Focus On Ontologies

Ontologies are instruments for structuring and encoding knowledge, thus enabling machines to comprehend and process it in an automated manner. To achieve this goal, ontologies are provided with the following basic elements:

Class. A class (also named entity) is an abstraction mechanism for grouping resources with similar characteristics. Classes are sets that can represent concrete categories, such as *Student* and *PersonalComputer*, and/or abstract categories, such as *Privacy* or *Compliance*.

Individual. An individual is an instance of a class and represents a single object. For example, *StudentA* could be an instance of the entity *Person*.

Property. A property (also named relation) describes how two entities relate. Such combination results in the ontological construct of *triple*. For example, the property *hasColleague* can be used to relate the two individuals *StudentA* and *StudentB*, which leads to the triple *StudentA hasColleague StudentB*.

Axiom: An axiom is a statement that is asserted to be true in the described domain. For example, *Student hasDevice only (PersonalComputer or Tablet* is an axiom asserting that any devices owned by students are either a *PersonalComputer* or a *Tablet*.

Reasoning: Reasoning is a fundamental component of ontology-based knowledge representation. It plays a crucial role by enhancing the intelligence and capabilities of systems that use ontologies to represent and reason about knowledge. At its core, ontological reasoning aims to make inferences, and answer queries, based on the knowledge encoded in an ontology. To understand ontological reasoning, it is essential to delve into the three key components: the *ABox*, *TBox*, and *RBox* [60]. In particular, the ontological entities and the properties belong respectively to TBox and RBox, while the facts about individuals belong to ABox.

In the context of ontological reasoning, the following benefits are gained:

- **Inference and Reasoning**: Ontological reasoning involves making logical deductions and inferences based on the information present in the ABox, TBox, and RBox.
- **Classification**: Classification is a critical aspect of ontological reasoning that assigns individuals to appropriate classes based on the TBox information and ABox assertions. This ensures that instances are correctly categorised within the ontology.
- **Consistency Verification**: One of the primary tasks in ontological reasoning is to verify the consistency of the knowledge base. Reasoning algorithms are employed to check whether the information in the ABox, TBox, and RBox is logically coherent. If inconsistencies are detected, they may signal errors in the ontology or the data.
- **Query Answering**: Ontological reasoning facilitates answering complex queries by leveraging the structural knowledge from the TBox, the factual information from the ABox, and the role descriptions in the RBox. This enables systems to respond to user queries with precise and context-aware results.

Differences between ontologies and classical approaches of representations, such as databases, are described by Sir et al. [124]. Here, we recall the three main differences which are suitable for our purposes: a) Ontologies enable semantic reasoning through inferences, c) Ontologies enable global sharing and integration of data (Linked Data), thus allowing users and applications to link different types of data coming from various sources. While databases are conceived for storing data, ontologies make knowledge more suitable for interoperability with systems and other ontologies. Such interoperability is crucial in our work since the security measures could be used by applications to automate the compliance verification and for extending the application scope of the proposed ontology; c) databases act in the Closed World Assumption (CWA), while ontologies act in the Open World Assumption (OWA). CWA is the presumption that a true statement is also known to be true; conversely, what is not currently known to be true is false. The opposite of CWA is OWA, which states that a lack of knowledge does not imply falsity.

The following example, which is loosely based on a lecture by Ian Horrocks, serves to emphasise such a difference. Written in the OWL syntax, the example assumes the creation of a general knowledge base for the management of university personnel by defining specific classes, properties, axioms, and individuals.

```
CLASSES
 Human, Professor, Student,
 Personal Computer, Tablet,
 Security Token

 PROPERTIES
 hasDevice, isDeviceOf,
 hasColleague

 AXIOMS
 Student hasDevice only
  (PersonalComputer or Tablet)

 hasDevice
  Inverse: isDeviceOf
  Range: Human

 SecurityToken isDeviceOf
 only Professor
```

```
INDIVIDUALS
StudentA (Type Student),
StudentB (Type Student),
StudentZ (Type Student)

ProfessorA (Type Professor),
SecurityTokenA (Type SecurityToken)

FACTS
StudentA hasColleague StudentB
StudentA hasDevice PC-A

SecurityTokenA isDeviceOf ProfessorA
```

Table 3.1: Key Differences between databases and ontologies in querying knowledge

#	Query/Operation	Database Response	Ontology Response
a.	Is *StudentZ* colleague of *StudentA*?	No	Unknown
b.	Getting number of colleagues	2	At least 2
c.	Inserting *ProfessorA* and *SecurityTokenA*	Error	Correct

It is worth noting that the output of the ontology is more reactive to new developments and more adaptive to the unknown. In scenario (c.), these observations regarding the various categories of assumptions appear to be more relevant. Even if Professor A's humanity is not explicitly demonstrated, ontologies assume that he is human and a professor because he possesses a security token. This allows Professor A to have a security token. Instead, databases are unable to produce these derivations because of CWA, which would have resulted in a constraint breach. Moreover, the information provided by item (c.) is helpful for our ontological representation goal, since it illustrates how inferences are essential to determining the precise security measures that apply to the referred agent.

3.4 The SecOnto Methodology

The proposed methodology for representing legal security documents through machine-understandable encoding fulfils the approaches proposed by Fenz et al. [48], Distinto [36], and Palmirani et al. [102]. The goal is to represent security directives via ontologies, regardless of the use of underlying encoding, by:

- describing in detail how and which semantic structures are more appropriate for the represented entities and relations, taking into account the nature of the legal security document and also paying particular attention to the grammar of security measures and their treatment;
- providing an agent-centric representation, respecting the nature of each security measure, each one associated with a specific agent that needs to execute it;
- adding specifications to the compliance and post-compliance verification activities, the last one consisting of SPARQL queries for differential analysis.

In what follows, we present the methodological steps for defining the semantic structure, and then we show how it is validated by its application on the NIS 2 Directive.

For the design of the SecOnto methodology, we took into account the renowned Methontology [50] methodology, which is well-established in the field of ontological development. Methontology consists of seven steps, namely 1) *Specification*, that provides semi-formal ontology specification documents; 2) *Knowledge Acquisition*, that collects all the knowledge required to build the ontology; 3) *Conceptualisation*, that provides a conceptual model of the ontology; 4) *Integration*, that considers the reuse of definitions already built into other ontologies; 5) *Implementation*, that produces the ontology; 6) *Evaluation*, that carries out a technical judgement of the ontology, and 7) *Documentation*, that produces all the documents explaining both the technical and ontological choices.

The SecOnto's methodology steps and the related outcomes are depicted in Figure 3.1.

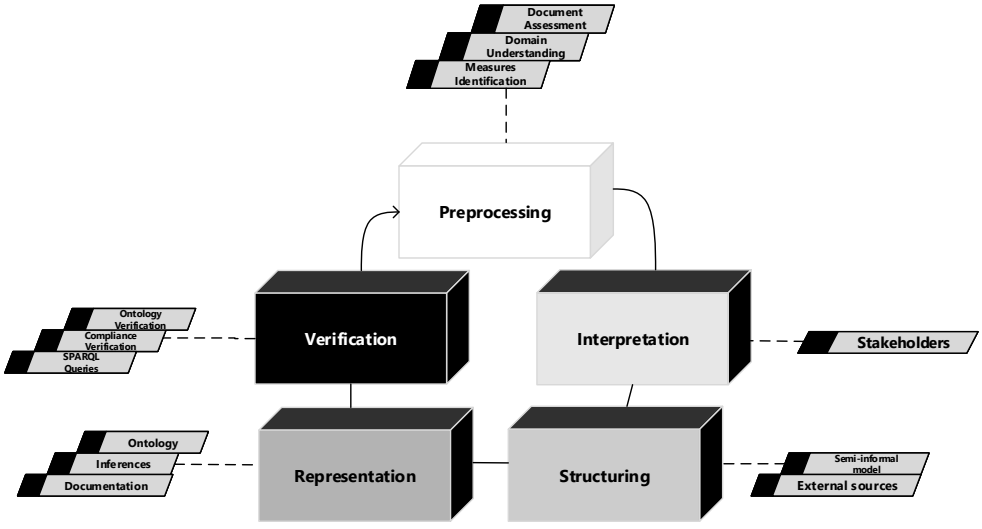


Figure 3.1: Overview of the SecOnto methodology

SecOnto reconsiders Methontology's steps in light of domain-specific considerations. Consequently, SecOnto foresees five steps.

In the first step of the SecOnto methodology, the Preprocessing step, a general assessment of the document and its content is made for the sake of individualising and evaluating the structure of the legal document and

the different ways the security measures are structured. The outcomes of the step include a general understanding of the document. It anticipates the first step of Methontology, the 1 Specification step, and partially overlaps with step 2, Knowledge Acquisition, since a comprehensive analysis of the legal documents is needed to understand the domain of knowledge fully and to acquire all the required information.

As a second step, `SecOnto` prescribes the Interpretation step for identifying the measures' components, such as the subjects, actions, and objects (which will compose the ontological triples). These are the primary outcomes of the step. It merges the Methontology steps 1 Specification and 2 Knowledge Acquisition.

The Structuring step of `SecOnto` merges the Methontology's steps 3 Conceptualisation and 4 Integration, since it provides the first semi-informal description of the ontological model and the required knowledge, which are also the primary outcomes.

Then, `SecOnto` provides the Representation step, where security measures are fully translated into the ontological model. This step corresponds to the Methontology step 5 Implementation. The primary outcome is the ontology describing the measures, the related documentation, and the logical inferences obtained from the ontology.

`SecOnto` reaches its conclusion with the Verification step, demonstrating the utilisation of the created structures for compliance verification through ontological reasoning, and leveraging SPARQL language for post-compliance verification differential analysis. This step includes the Methontology step 6 Evaluation, integrated with the compliance verification of the measures. The outcomes of the step are the ontology verification, the SPARQL queries verifying the compliance of the measures and the compliance verification itself.

Each step is composed of several micro-steps, for which we provide a brief introduction, the description, and the validation of the micro-step through a real-world case study. When required, we offer some additional validation cases to explain the purpose of the micro-step better.

Each step is conceived to be executed manually, because it is impracticable to be carried out automatically, due to its highly accurate, and sometimes subjective, nature. In the section *Tool support*, however, we show how semi-automatic approaches can assist the accomplishment of each step, together with their limitations and challenges.

3.4.1 Step 1: Preprocessing

The first step of the SecOnto methodology is Preprocessing, a preliminary step required for the general assessment of the legal documents and the related content. It is usually carried out manually and is focused on the parsing of the document to understand the overall structure of the document, for example, whether it provides a set of premises and a list of articles, recognising how many items constitute each article, and pointing out the structural hierarchies of each item. The Preprocessing step features three inner micro-steps.

3.4.1.1 Identification of articles

The document articles expressing relevant security measures are identified.

Description The aim is to identify the articles that describe the measures helpful in improving the overall security posture of European Member States and organisations and the responsible agents of such measures. Therefore, articles that describe content of other nature, such as abbreviations and general provisions, are ignored in this step.

Validation To validate the step through the NIS 2 Directive, we take into account the articles containing the most suitable security measures for the ontological representation, namely, the articles from 7 to 37, within the chapters from II to VII.

3.4.1.2 Identification of article items

The document articles items that provide the greater granularity in terms of measures items are identified.

Description The selection of article items is strictly related to the specific objectives and characteristics of the analysis; hence, the selection criteria should be adapted accordingly. However, we note that the articles selected by the previous step may consist of multiple paragraphs and sentences, and the items must be granular, to obtain separated security measures, each one composed of subjects, actions, objects and any other relevant ones.

Validation

Validation 1 To validate the step, we take into account paragraph 3 of Article 7 of the NIS 2 Directive. We identify:

> Member States shall notify their national cybersecurity strategies to the Commission within three months of their adoption. Member States may exclude information which relates to their national security from such notifications.

Which can be split into two items:

> 1 - Member States shall notify their national cybersecurity strategies to the Commission within three months of their adoption.
>
> 2 - Member States may exclude information which relates to their national security from such notifications.

Validation 2 Now, consider paragraph 6 of Article 28 of the NIS 2 Directive, we identify:

> Compliance with the obligations laid down in paragraphs 1 to 5 shall not result in a duplication of collecting domain name registration data. For this purpose, Member States shall require TLD name registries and entities providing domain name registration services to cooperate with each other.

which can be split into two items:

> 1 - Compliance with the obligations laid down in paragraphs 1 to 5 shall not result in a duplication of collecting domain name registration data.
>
> 2 - To that end, Member States shall require TLD name registries and entities providing domain name registration services to cooperate with each other.

3.4.1.3 Reassemble of articles items

The document articles items are reassembled to obtain consistent and non-overlapping security measures.

Description The measures may be overlapped across multiple items. Some articles may feature items that are not self-contained, but develop into sub-items expressing multiple objects of the security measure. Hence, to reduce the risk of inconsistencies and preserve the overall coherency and accuracy of the resulting representation, we need to: a) extract subjects and actions from super-items, b) extract the objects from the sub-items, c) reassemble them, and d) link the sub-items with the super-items.

Validation

Validation 1 Let us consider paragraph 3 of Article 23 of the NIS 2 Directive. We can reassemble the following two items:

> An incident shall be considered to be significant if: (a) it has caused or is capable of causing severe operational disruption of the services or financial loss for the entity concerned; (b) it has affected or is capable of affecting other natural or legal persons by causing considerable material or non-material damage.

as follows:

> An incident shall be considered to be significant if it has caused or is capable of causing severe operational disruption of the services or financial loss for the entity concerned;
>
> An incident shall be considered to be significant if it has affected or is capable of affecting other natural or legal persons by causing considerable material or non-material damage.

Validation 2 Consider paragraph 2 of Article 27 of the NIS 2 Directive. We can reassemble the following items:

> Member States shall require entities referred to in paragraph 1 to submit the following information to the competent authorities by 17 January 2025: (a) the name of the entity; (b)

> the relevant sector, subsector and type of entity referred to in Annex I or II, where applicable; (c) the address of the entity's main establishment and its other legal establishments in the Union or, if not established in the Union, of its representative designated pursuant to Article 26(3); ...

as follows:

> Member States shall require entities referred to in paragraph 1 to submit to the competent authorities by 17 January 2025 the name of the entity;
>
> Member States shall require entities referred to in paragraph 1 to submit to the competent authorities by 17 January 2025 the relevant sector, subsector and type of entity referred to in Annex I or II, where applicable;
>
> Member States shall require entities referred to in paragraph 1 to submit to the competent authorities by 17 January 2025 the address of the entity's main establishment and its other legal establishments in the Union or, if not established in the Union, of its representative designated pursuant to Article 26(3);
>
> ...

3.4.2 Step 2: Interpretation

Once articles and items have been identified, an a priori analysis of the security measures is necessary to interpret their components correctly. The Interpretation step carries out three interpretive micro-steps, including the examination and understanding of not only the relevant grammatical elements but also of the semantic structures that constitute a security measure. Both the automated and manual approaches may be used for this step: the first one using NLP libraries to extract the parts of speech, and the second one manually analysing each measure.

The step aims to anticipate a defined representation that captures the semantics of the security measure.

3.4.2.1 Interpretation of entities

The security measures entities are interpreted to associate them with ontological entities.

Description The first step of Interpretation consists of a detailed examination of the Directive, starting with the interpretation of *Directive Entities* (DEs). This is crucial for the subsequent definition of ontological entities (OEs), which are pivotal components of ontology development. In the context of European directives, we may find a single article dedicated to listing some DEs, preceding the subsequent articles where the measures are expressed, or find more specific DEs defined within the measures themselves.

Validation

Validation 1 Since we usually deal with directives governing information technology, we identify at least two types of DEs, namely, person-related and technology-related DEs. Without loss of generality, we can assume the existence of a super-class, called *Actor*. By following the OWA principles, articles listing the DEs are not complete and exhaustive. This suggests that the DEs, which will be subsequently mapped into OEs, are built together with the analysis of each article.

Assuming the existence of the OE *Actor*, we specialise it and create sub-classes for distinguishing between human-centred and technological-centred DEs, respectively mapped by OE *Agent* and *System*. The OE *Agent* includes both the human-centred DEs involved in the security measures and all the DEs that express actions, for example, the NIS 2 CSIRT. Additionally, the OE *Agent* includes the human-centred actors even though they play a passive role in some measures, for example, in the NIS 2 directive, the CSIRT is designed by a *Member State*. The reason for such a choice follows the intrinsic nature of the NIS 2 Directive, which is a best practice for someone who must take action to comply; hence, we convey that those entities can assume both passive and active roles.

Validation 2 In the context of the Directive, technology-oriented entities also exist, which should be interpreted differently. The decision to define role-less entities does not impact the model's generality. Systems could be passively used by users and by other entities such as search engines or online marketplaces. Still, they are the recipients of the

protection pursued by the security measure. At the same time, a system could be an active tool. The NIS 2 does not explicitly refer to the tools used in cyber-kill chains of vulnerability assessment and penetration testing activities. Still, such activities are beneficial for demonstrating the robustness of systems and organisations. The same idea applies to all infrastructures created for security testing: they could be generally identified as a system and specialised accordingly.

3.4.2.2 Interpretation of agents and actions

The security measure agents and actions are interpreted to identify the related ontological roles.

Description The measures contained in each article of the document have a dual purpose: establishing actions that the subject must accomplish and providing additional specifications for the DEs. Each article is associated with at least one specific subject, although intricate articles may involve multiple subjects. Then, the agents and related actions within a security measure must be interpreted. By systematically analysing the measures, we ultimately break down the sentences within each article through the interpretation process applied to the subject (agent) and the verb (action). This deconstruction process greatly simplifies the interpretation phase.

Validation

Validation 1 To provide a first validation of this micro-step through the NIS 2, we quote an excerpt from paragraph 1 of Article 7:

> Each Member State shall adopt a national security strategy that provides for the strategic objectives, the resources required to achieve those objectives, and appropriate policy and regulatory measures, with a view to achieving and maintaining a high level of security. ...

Since the focus of the measure is the subject Member State, we can interpret it as an *Agent* OE. Once the agent of a measure has been interpreted, we identify the actions that such agent must undertake. In our example, we can interpret *Adopt* as the action to be performed by the subject Member State.

Validation 2 To provide a second validation of this micro-step through the NIS 2, we quote an excerpt from paragraph 4 of Article 11:

> The CSIRTs shall establish cooperation relationships with relevant stakeholders in the private sector to achieve the objectives of this Directive.

Since the focus of the measure is the subject *CSIRT*, we can interpret it as an Agent OE. Furthermore, we can interpret *Establish* as the action to be performed by the subject CSIRT.

3.4.2.3 Interpretation of objects and complex sentences

The security measure objects and complex sentences are interpreted to identify the related ontological roles.

Description Agents and security measure actions were interpreted in the preceding micro-step, but to complete an ontological triple, the interpretation of the objects must be addressed. In the most complicated circumstances, the objects are not atomic and must be extracted from complex sentences. When a security measure does not follow the standard structure consisting of subject, action, and object, we can apply two different strategies: a) splitting the sentence into short and simple clauses, and b) assuming that the part of the sentence following the object qualifies the object itself.

Validation

Validation 1 In the validation 1 of the previous step, we already interpreted the couple *Member State* and *Adopt* as the subject and action of the security measure, respectively, but the interpretation of the object and of the sentence is still missing. In the NIS 2 Directive, we can assume that the scope of a single verb is limited to a sentence. Then, we can easily identify the phrase *national security strategy* as the object of the measure. By combining all the elements interpreted so far, we can deconstruct the measure as the triple *Member State Adopt National Security Strategy*.

Since the excerpt of the previous step is a complex one, we must decide one of the strategies as mentioned earlier; the first strategy produces two triples, a) *Member State Adopt National security Strategy* and b) *National Security Strategy Includes strategic objectives, the resources required...*, whereas the second approach produces b1) *Member State Adopt*

National Security Strategy, b2) *National Security Strategy has as qualifying DEs Strategic Objectives* and b3) *National Security Strategy has as qualifying DEs Regulatory Measures*, etc. Since those entities are too complex to qualify, we believe the first strategy to be more suitable.

Validation 2 In the validation 2 of the previous step, we already interpreted the couple *CSIRT* and *Establish* as the subject and action of the security measure, respectively. We can easily identify the phrase *Cooperation Relationships* as the object of the measure. By combining all the elements interpreted so far, we can deconstruct the measure as the triple *CSIRT Establish Cooperation Relationships*. Since the excerpt of the previous step is a complex one, we approach the first strategy, producing two triples; a) *CSIRT Establish Cooperation Relationships* and b) *Cooperation Relationships Established With Relevant stakeholders in the private Sector*.

The period "with a view to achieving the objectives of this Directive" does not add anything significant about the agent's actions, so we can decide to exclude it from the future ontological representation.

3.4.3 Step 3: Structuring

The Preprocessing and Interpretation steps trace the path for the ontological representation of a security document and the related security measures. However, they inherently lack the structured foundation required for knowledge representation, while it is clear that a semantic design is a crucial prerequisite for the definition of the ontology. By establishing a semantic design, we strengthen the precision in representing the entities and relations within the directives, hence the need for the Structuring step.

The Interpretation step delivers a set of DEs on which all security measures revolve, which can be further specialised. Moreover, it is necessary to keep the structure of the document. For this reason, the Structuring step is organised in two micro-steps to address the issues mentioned above, which can also anticipate some ontological terms.

3.4.3.1 Provision of an agent-oriented design

An agent-oriented design is sketched to provide an agent-centric structure to the ontology.

Description We observed that the measures are centred around a particular class of entities, namely *agents*. Thus, it would make sense to offer

an agent-oriented structure with the primary goal of making it easier to find measures in the ontology that are specifically associated with that particular agent.

Validation Considering that the compliance verification regards one of the possible agents named in the security directives, the first element in our design is dedicated to such entity, here conceptualised as *EntityX*, for example, a NIS 2 Member State or a GDPR Data Protection Officer (DPO). Then, we represent each measure through a specific entity, let us call it *Article-Y-Entity-X-Measure*, that is endowed with all the definitions providing the compliance requirements for the related measure. Next, we represent the article that endows all the previously constructed measures, namely, *Article-Y-Entity-X-Compliant*. The latter entity is related to the union of endowed measures through an equivalence relationship, since the compliance is verified if and only if all the measures turn out to be compliant at the same time. Moreover, *Article-Y-Entity-X-Compliant* is also of type *Article-Y* since each article provides different compliance directives for different agents. Finally, each *EntityX*, for which compliance must be guaranteed, is related to *Article-Y-Entity-X-Compliant* through an equivalence relationship, thus any contradiction can be interpreted as the absence of compliance. These considerations are summarised in Figure 3.2.

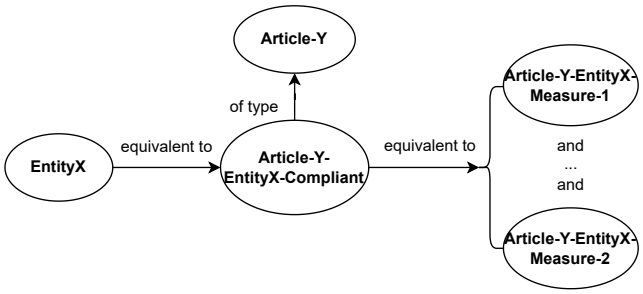


Figure 3.2: Overview of the main classes for the security directives

For example, Article 7 of the NIS 2 Directive involves CSIRT (Computer Security Incident Response Team), and consequently, by following such design, we can assume that *Article10* includes both *Article-10-MemberState-Compliant* and *Article10-CSIRT-Compliant*. Concerning the first entity, the specification of the related measure is represented by *Article10-MemberState-Measure*, connected to *Article10-CSIRT-Compliant* through an equivalence relationship. *Article10-MemberState-Measure* de-

scribes the compliance measure, for instance:

$$Article10-MemberState-Measure \equiv \exists adopt.NationalCybersecurityStrategy$$

Let us suppose to introduce *EntityX-test* as a member of *MemberState*. If and only if *EntityX-test* satisfies all the definitions provided by *Article-10-MemberState-Compliant*, hence it satisfies all the definitions provided by *Article10-MemberState-Measure*, we can infer the *EntityX-test* is a *MemberState*, otherwise we obtain a contradiction.

By adopting this type of design, we can represent the security directives with small and separated components, facilitating the management of the ontology and the directives themselves. The separation between entities and their respective measures aims at a loose coupling in support of readiness for modifications if any errors are found or modifications become necessary.

3.4.3.2 Provision of a document-oriented design

A document-oriented design is sketched to preserve the legal document structure.

Description An in-depth analysis of a legal document shows that an agent-oriented design may not be enough, since some measures directly refer to specific parts of the text, suggesting that the structure of the document has its relevance in the realisation of the ontology and must be kept.

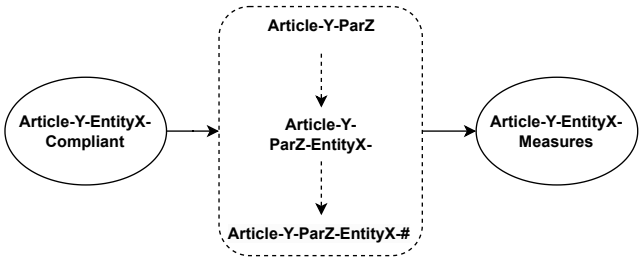


Figure 3.3: Representation of specific paragraphs

Validation The document-oriented design further characterises the output of the previous step, as shown in Figure 3.3.

Leveraging such a design, we integrate an additional element between the one representing the compliance within an agent and the measures related to the specific article. In the new element, the specific structure of the document is reflected, and thus, the measures related to the particular entity are reflected. The feature we add with this integration is essentially the navigability of the measures. If the agent-oriented design logically represents compliance, the document design moves towards the definition of the ontology, adding the entities for representing the specific paragraphs and the items (as conjectured in the Preprocessing step), the paragraphs related to an agent (or multiple agents), and, if any sub-lists exist, sub-actions or their further specification.

This approach helps to directly reference a paragraph (or item of sub-list, indicated in the previous figure by the number sign) of a specific article and maintain the coherent structure of the document.

3.4.4 Step 4: Representation

The Interpretation and Structuring steps lay the foundation for ontological construction, allowing us to capture the fundamental elements of a measure and the internal role of the characterising elements, and to give them a semantic design, respectively. The semantic design aims to abstract the "external relational structures" since they suggest how the relevant entities (the agents) can be globally connected to the associated security measures. Notably, we do not always face simple sentences consisting exclusively of the main part-of-speech (POS) tags (subject, verb, object). By contrast, there are often very complex structures, due to the legal nature of the document, where the main agent is still the POS subject, the main action is still the POS verb, but the object opens up further possible levels of characterisation of the actions themselves. Arguably, these need to be inspected too. Therefore, there are three micro-steps here to address structural and grammatical insights and provide a general solution for the final representation of a measure.

3.4.4.1 Management of structural insights

The intricacies of managing structural insights within the document are analysed.

Description As highlighted during the Preprocessing step, a single security measure is expressed within a single sentence that we identified as an item, which is, in turn, contained inside a paragraph. An article

may be composed of at least a paragraph, sometimes identified within a number, sometimes not. A single security measure can also be developed in-depth, with hierarchies depth of at most 2. Since these articles are frequently referenced in other articles, the measures about their article of membership are not likely to be directly correlated.

Validation The following scenarios of structural insights may arise in when dealing the European security directives.

Scenario 1: Multiple depth levels Two cases exist: single depth level, which could contain specific actions to perform or just the objects of the action expressed by the previous parent node; double depth level, which always contains the objects of the action expressed by the previous parent node. This is the case of paragraph 3 from Article 15 where the tasks of the CSIRTs network are distributed among two levels of specifications; the first level for expressing the operations, and a second level about the characterisation of the task (task of point j).

Scenario 2: Modal verbs Given the forward-looking nature of the application of measures in a legal document, the verbs are often complemented by the use of modal verbs. Following the precedents of several European regulations, two modal verbs in particular stand out: *shall* and *may*. The modal verb *shall* plays a prominent role in almost all security measures. This prevalence is because it effectively expresses the future obligation to comply with certain regulations. In essence, *shall* not only conveys a sense of impending responsibility but also serves as a definitive indicator of ordered courses of action. It is the cornerstone of legal language that ensures a sense of duty and compliance with regulatory standards. Conversely, the modal verb *may* is used much less frequently in the context of security measures. *May* means the possibility of taking a certain action, which sets a more permissive and discreet tone in legal texts. It is a rarity in this area, reflecting the cautious approach of allowing flexibility and choice within the prescribed security procedures. When the verb *may* appears, it often represents an exception rather than the rule and highlights the importance of compliance with certain regulations and the limited circumstances in which flexibility is permitted.

Therefore, a strategy for representing the verbs and thus the ontological object properties should take into account how many repetitions of such modal verbs can, firstly, overload the reading of the ontology and, secondly, represent useless repetitions. Under this assumption, we

suggest representing the verbs supported by *shall* in their natural form, without explicit reference to shall, while the verbs supported by may are written in the form *may+Verb*, to make the appropriate distinction.

This is the case of paragraph 7 from Article 10 ("The CSIRTs may establish cooperation relationships with third countries ..."; in the ontological representation, *establish* becomes *mayEstablish*, in order to be distinguished by the eventual shall-case.

Scenario 3: Passive verbs We treat the measures containing a passive verb (so a passive action) as the active measures, i.e. by identifying the POS. This means that the triples we create in the passive case reflect the same structure.

Taking into account paragraph 2 of Article 16 ("EU-CyCLONe shall be composed of the representatives of Member States"), the triple agent, action, object becomes: EU-Cyclone (agent) ComposedOf (action) Representatives of Member States (object).

Scenario 4: Expressing specific values The representation of measures requires the integration of data properties that are closely linked to certain values. This need is particularly evident when it comes to numerical data, such as economic amounts, time intervals or other quantitative information. These data properties serve as the basis for precise and context-rich ontological representations and enable comprehensive understanding and analysis of measures related to more administrative aspects. However, when it comes to dealing with dates, the complexity of the task can often exceed the capabilities of the produced ontology, given the complexity and variety of date-related information. In such scenarios, integrating an external ontology becomes not just an option but a necessity. External ontologies can provide the semantic richness and depth required to effectively model and capture the peculiarities of dates and related temporal information. This broader integration ensures that the ontology is equipped to handle various temporal complexities such as historical context, date formatting, time zones and recurring data, which are critical to the accurate representation and interpretation of measures.

A clear example is present in paragraph 3 of Article 7, where a specific value (3) is needed to express the time interval for the adoption of the national cybersecurity strategy.

Scenario 5: Compacting complex objects Creating such an ontology involves some complications arising from the nested semantic structures and individual entities used to represent what we already called complex objects. In fact, when translating any security measure, there is a need to create ad hoc entities that represent consistently and clearly the associated object. Creating triples that include subjects, verbs, and objects poses an additional challenge when it comes to generating the corresponding ontological Internationalised Resource Identifiers (IRIs). This challenge arises from the fact that in most cases, not every part of speech can be summarised in just a few words. It is worth noting that only a limited number of words, typically no more than three to four words at most, meet this concise criterion, a determination based on empirical evaluation.

Two approaches can be used: the first one concerns the subjects and objects, and the second one the properties, i.e. the verbs. In the first approach, we exclude every article, prepositions, and the other grammatical conjunctions that may overload the IRI conciseness. It is usually more complicated to deal with the object than the subject. In many cases, the agent typically consists of a series of concise words, which simplifies its representation without compromising its semantic richness. Conversely, objects tend to be more complex and require a more peculiar approach that alludes to and exploits the genitive possession. Though the meaning of each measure is clear (only one interpretation is possible), writing IRIs depends strictly on ontology engineering, which makes it possible to assign more weight to certain words and thus place them entirely subjectively. Worth noticing, however, that such decisions may strictly depend on the vision of the modeller. As an illustrative case for IRI formulation, let us consider a hypothetical scenario with a complete object structured as *A B of C*, where A, B and C represent nouns and *of* acts as a grammatical preposition. Using a form similar to the possessive genitive, we can consider creating an IRI such as *CAB* to associate with the above object.

This small case illustrates a general solution that is recursively applicable when the complexity of the object rises, but which must be assessed individually, under the present nouns and verbs that make up the object. From Article 11, paragraph 5, we have the following measure: "the CSIRTs shall promote the adoption and use of common or standardised practices". The object *use of common practices* becomes *Common Practices Use*.

In the second approach, it could only represent the verb itself, in particular when the relation between verb and object is transitive. In such

a case, the corresponding object property reflects the verb. In the hardest case, when the relation is intransitive, the object must be integrated with prepositions or grammatical conjunctions, to give a fluid meaning to the report, as if one were reading the directive. Considering again paragraph 5 of Article 5, with this second approach we have: *promote Adoption Of* as the action, and *Common Practices* as the object.

Note how these considerations are strictly related to the ones made up in the third micro-step of the Interpretation step; in the second case we are just interpreting at high-level the agents and the related objects, but in current first case, we have already interpreted the structure of the triple, so the agent, and we are considering how managing the encountered complex objects.

These succinct examples show general solutions that can be applied recursively to handle the increasing complexity of an object. However, it is important to emphasise that these approaches must be adapted on a case-by-case basis, taking into account the specific nouns and verbs that make up the object.

Scenario 6: Unnumbered paragraphs In the document, some articles' paragraphs may not be numbered, making it unclear whether they belong to the previous one. In this case, we distinguish them from the others thanks to the root *Unc*, followed by the cardinal number of the uncategorised paragraph.

Throughout the document, several instances of this structural insight exist: in Article 11, between the numbered paragraphs 3 and 4, we can find two of these cases.

Scenario 7: Hierarchical restrictions A single security measure that falls within the context of a single sentence belonging to a paragraph (which may or may not contain possible sublists) is represented according to the following proposed schema.

```
{''action-1'': ''object-1'' and ''action-2'' : ''object-2'', and {''action-
2-1'': ''object-2-1'', and ''action-2-2'': ''object-2-2'', ... { ... }
and ''action-2-j'': ''object-2-j'', } ... ''action-i'': ''object-i'', } and
{ ... } }
```

The adoption of this schema is effective when complex security measures are in place at multiple hierarchical levels.

The aforementioned hierarchical levels are evident when organising a complex measure and defining the goal of a particular activity. Such a situation occurs when the object recursively opens sub-clauses that give some of the object's distinguishing characteristics and that can be stated using triples.

As a result, the suggested technique gets beyond the ontologies' language syntax constraints, as there are no particular logical constructs that may be used to characterise objects in greater detail.

Handling such a structural insight with the logical conjunctions as described will be further illustrated in the next steps of Representation and Verification.

3.4.4.2 Exemplification of ontological representation

An exemplification of the ontological representation according to the previous methodological steps is provided.

Description Considering all the assumptions carried out from the first step of the methodology up to the last illustrated step, the ontology representation is exemplified.

Validation

Validation 1 Taking into account an excerpt from paragraph 1 of Article 7, its ontological representation within OWL syntax is:

```
CLASSES
MemberState, NationalSecurityStrategy,
Article7, Art7Par1-MemberState, Art7Par1-NCS,
Article7-MemberState-Compliant, Article7-NCS-Compliant

PROPERTIES
adopt, include

AXIOMS
Art7Par1-MemberState subclassOf Article7
Art7Par1-NCS subclassOf Article7

Art7Par1 EquivalentTo
 MemberState and (adopt some NationalSecurityStrategy)

Art7Par1 EquivalentTo
 NationalSecurityStrategy and (include
```

```
 some SecurityPolicy)

Article7-MemberState-Compliant EquivalentTo Art7Par1-MemberState

Article7-NCS-Compliant EquivalentTo Art7Par1-NCS
```

The ontological representation reflects the considerations made in steps 1-4 of SecOnto methodology, and in particular: the subdivision of the articles into single items deriving from Preprocessing step (e.g., Art7Par1); the use of the POS tags as ontological entities and properties (e.g., *MemberState* and *provide*, respectively) deriving from the Interpretation step 2; the axioms structured as the provided design in Structuring step; and the additional grammatical and structural considerations provided in this Representation step.

Validation 2 Taking into account an excerpt from paragraph 2 of Article 12, its ontological representation within OWL syntax is:

```
CLASSES
ENISA, EuropeanVulnerabilityDatabase,
Article12, Art12Par2-ENISA,
Article12-ENISA-Compliant

PROPERTIES
develop, maintain

AXIOMS
Art12Par2-ENISA subclassOf Article12

Art12Par2-ENISA EquivalentTo
 ENISA and (develop some EuropeanVulnerabilityDatabase and
 maintain some EuropeanVulnerabilityDatabase)

Article12-ENISA-Compliant EquivalentTo Art12Par2-ENISA
```

Also in this case, the classes have been named according to the structures foreseen by the previous steps, while the axioms provide a logically valid representation of the security measures.

3.4.5 Step 5: Verification

The task of compliance verification is notoriously complex. It is usually done via auditing between operators, hence often causing inconsistencies.

So, if we use an automated and time-saver instrument such as the reasoning, it would reduce time and related costs. Ontologies, as mentioned, meet those prerogatives.

Once such security measures are encoded in a logically deductible way, the current step of verification can be automatically performed thanks to automated reasoning.

This step, which is composed of three inner micro-steps presented in the next subsections, leverages the following exemplary knowledge base where, in the context of NIS 2 Directive, as individuals, the entities agents, relations and objects, can assume the following values:

$\texttt{Articles} = \{\texttt{AR1}, \texttt{AR2}, \texttt{AR3}, \ldots\}$
$\texttt{Agents} = \{\texttt{MemberState}, \texttt{CSIRT}, \texttt{ENISA}, \ldots\}$
$\texttt{Relations} = \{\texttt{adopt}, \texttt{designate}, \texttt{assess}, \ldots\}$
$\texttt{Objects} = \{\texttt{Obligations}, \texttt{NCS}, \texttt{EVD}, \ldots\}$

3.4.5.1 Instantiating individuals

The ontological individuals are instantiated to cover some compliance cases.

Description For the instantiation of individuals, an agent could be taken into account, and for example, a *MemberState* or *ENISA*, which, thanks to the design provided, have strong influences throughout the methodology. Then, some measures to which the agent adheres are associated with them. The measures will be next evaluated in the reasoning phase towards compliance verification. In the end, the instantiating of the individuals may be used to construct particular SWRL rules and then further enforce the security measures.

Validation

Validation 1 Assume that the articles express the following, hypothetical, measures, whose compliance should be verified:

- $\mathcal{AR}_1$: Member States shall designate the national security strategy and adopt prescribed obligations.
- $\mathcal{AR}_2$: Member States shall assess the national security strategy.

which translate to the individuals (I):

- $\mathcal{I}_1$, which owns all security measures of both articles;

$\mathcal{I}_1 = \{\texttt{designate(NCS)}, \texttt{adopt(Obligations)}, \texttt{assess(NCS)}\}$

- $\mathcal{I}_2$, which owns the security measures of the first article;
  $\mathcal{I}_2 = (\mathcal{I}_1 \setminus \{\texttt{assess(NCS)}\}$
- $\mathcal{I}_3$, which owns only some security measures.
  $\mathcal{I}_3 = (\mathcal{I}_1 \setminus \{\texttt{designate(NCS)}\} \cup \mathcal{I}_2 \setminus \{\texttt{assess(NCS)}\}$

Validation 2 Furthermore, assume the following measures:

- $\mathcal{AR}_3$: ENISA shall develop the European vulnerability database.
- $\mathcal{AR}_4$: ENISA shall ensure the integrity of the European vulnerability database.

which translate on the individuals:

- $\mathcal{I}_4$, which owns all security measures of both articles;
  $\mathcal{I}_4 = \{\texttt{develop(EVD)}, \texttt{ensureIntegrityOf(EVD)}\}$
- $\mathcal{I}_5$, which owns the security measure of the fourth article;
  $\mathcal{I}_5 = (\mathcal{I}_4 \setminus \{\texttt{develop(EVD)}\}$

3.4.5.2 Execution of Reasoning

The reasoning is executed to logically derive the individuals' compliance to the security measures.

Description Once executed the reasoner, the inferences are carried out according to the model prescribed and measures related to instantiated individuals. The measures that the prescribed agent complies with will be the outcome, and consequently the typology of each individual. This verification approach proves particularly advantageous, especially in real-world scenarios, as it enables the simultaneous validation of the compliance of numerous individuals concerning the directive with just one reasoning execution, as opposed to running multiple tools separately.

Validation

Validation 1 Consider the following inferences:

$$\mathtt{I}_1 \models \mathtt{type}(\mathtt{MemberState}) \qquad \mathtt{I}_2 \models \mathtt{type}(\mathcal{AR}_1) \qquad \mathtt{I}_3 \models \mathtt{type}(\emptyset)$$

Regarding I_1, it is accurately inferred as an instance of MemberState because it satisfies all the security measures. In contrast, I_2 is inferred as AR_1 since it only complies with Article 1, failing to meet the requirements of AR2, and consequently, it cannot be inferred as a Member State. Lastly, I_3 is inferred as an empty set, signifying that it does not encompass measures from either of the articles. This inference-driven approach demonstrates its effectiveness by enabling us to examine simultaneously both the compliant instances and the not-compliant ones.

Validation 2 Consider the following inferences:

$$\mathtt{I}_4 \models \mathtt{type}(\mathtt{ENISA}) \qquad \mathtt{I}_5 \models \mathtt{type}(\mathcal{AR}_4)$$

Regarding I_4, it is accurately inferred as an instance of ENISA because it satisfies all the security measures of the agent ENISA. In contrast, I_5 is only inferred as AR_4 since it only complies with Article 4, failing to meet the requirements of AR3, and consequently, it cannot be totally inferred as ENISA.

3.4.5.3 Execution of Differential Analysis

The differential analysis is executed to derive the missing security measures.

Description The reasoning step carries out the logical inferences, and then to identify missing rules that make it possible to verify (non-)compliance, the differential checks by querying the knowledge base with SPARQL queries must be addressed.

Validation The following query allows us to confirm what the individuals are missing to achieve compliance and possess all MemberState or ENISA (both individuated in the query through the variable **Agent**) peculiarities.

```
SELECT ?article ?action ?object
   WHERE{
      nis:Agent owl:equivalentClass ?a .
      ?a owl:intersectionOf ?b .
      ?b rdf:rest* ?c .
      ?c rdf:first ?article .
      ?article owl:equivalentClass ?e.
      ?e owl:intersectionOf ?f .
      ?f rdf:rest* ?t .
      ?t rdf:first ?s .
      ?s owl:onProperty ?action .
      ?s owl:someValuesFrom ?object .
     MINUS {
      nis:Ind ?action ?objInd .
      ?objInd rdf:type ?object .
     }
   }
```

Applying the query to the illustrative knowledge base from Validation 1 will output what is missing by the respecting individuals.

$$\mathtt{I}_1 \models \mathtt{missing}(\emptyset)$$

$$\mathtt{I}_2 \models \mathtt{missing}(\mathcal{AR}_2)$$

$$\mathtt{I}_3 \models \mathtt{missing}(\mathtt{designate}(\mathtt{NCS});\mathtt{assess}(\mathtt{NCS}))$$

While applying the same query to the illustrative knowledge base from Validation 2 will output the following inferences.

$$\mathtt{I}_4 \models \mathtt{missing}(\emptyset)$$

$$\mathtt{I}_5 \models \mathtt{missing}(\mathcal{AR}_3)$$

In both cases, the keyword *missing* indicates what is missed by the individuals when applying the query of differential analysis. The query exemplifies how a proper differential analysis can be conducted and how extremely simple interpreting the results is, despite traditional and much more complex document consultation.

3.5 Evaluation And Validation Of SecOnto

We are now ready to focus our attention on the evaluation and validation of the methodology.

SecOnto is a comprehensive methodology developed for the ontological representation of security directives, offering a structured approach to translating legal security documents into an ontological representation. This process enhances the clarity and interoperability of security policies. However, a significant challenge arises from the current lack of established research providing criteria for evaluating methodologies within the field of ontologies. Consequently, we are unable to formally evaluate SecOnto as a methodology or measure its effectiveness using standardised metrics. Nonetheless, we can evaluate the ontologies produced by SecOnto using established structural metrics. These metrics assess various elements, such as the number of classes, attributes, and logical axioms, providing a quantitative measure of the ontology's complexity and comprehensiveness. However, these metrics alone do not capture the methodological effectiveness of SecOnto, apart from the possible purely structural choices.

The validation of SecOnto's effectiveness can instead rely on qualitative criteria, focusing on two primary aspects: the robustness of its foundational principles and the tangible outcomes it generates. SecOnto is grounded in and extends upon well-established research in the domain of ontological representation, particularly concerning policy representation, as discussed in the related work section. This foundation ensures that SecOnto inherits proven methodologies, which are further enhanced and tailored for specific use cases. A key strength of SecOnto lies in its ability to leverage ontological reasoning, a well-established tool within the field, to facilitate compliance verification. The methodology not only utilises reasoning to ensure the logical consistency of security policies, but also structures step by step the results of these reasoning processes in a way that simplifies compliance verification tasks. This structuring is particularly beneficial in complex scenarios where specific understanding and detailed documentation of security compliance are required. Therefore, the correctness of the obtained results acts as a self-validation of the methodology.

While various metrics are available for evaluating ontological constructs, these do not directly measure the effectiveness of SecOnto as a methodology. However, the structured approach of SecOnto offers qualitative benefits, such as improved clarity and organisation of security

policies, which are critical in ensuring comprehensive compliance.

Despite its strengths, SecOnto, like any advanced methodology, presents certain challenges. One of them is the meticulous nature of the verification process required at each stage, ensuring that security measures are accurately represented in the ontology according to predefined standards. Although this process can be seen as resource-intensive, it is a necessary overhead to guarantee the reliability and accuracy of the final compliance verification. This thoroughness is indispensable, particularly in sensitive fields, where errors in security representation can have significant consequences.

In conclusion, we can state that, while the lack of specific metrics currently constrains the direct quantitative validation of SecOnto as a methodology, SecOnto can be validated through qualitative assessments of its foundational integrity and practical outcomes, as illustrated step by step.

3.6 Automating SecOnto

Automating step 1 Concerning the first micro-step, it is challenging to establish an automatic heuristic capable of accurately selecting only the articles, and consequently the chapters, containing significant security measures, given the numerous security documents to which SecOnto can be potentially applied. For instance, we can take into account the GDPR and the NIS 2. Both documents share the same structure, but it is very hard to select the appropriate privacy and security measures automatically. However, any NLP library can be adopted to preprocess the text described in the second and third micro-steps. Moreover, suppose the second micro-step does not require any particular assumption that needs to be made. In that case, we can foresee employing an automatic tool for the third micro-step to associate the various sub-items with the correct subject from the super-item and reassemble the sub-items coherently within the super-items.

Automating step 2 As described, the main task of the Interpretation step is based on extracting the various components of the security measures. Also, in this case, we can leverage the NLP libraries such as SpaCy and ClausIE to extract the parts of speech. NLP techniques provide huge benefits if the document it mainly constituted by simple sentences but are inefficient with documents, such as the NIS 2, that are predominantly constituted by complex sentences. The Interpretation step seems the most

suitable for NLP techniques. However, as confirmed by the experiments already illustrated in Chapter 2, NLP tools do not extract the correct part of speech, at least in most cases. Therefore, in a sensitive context such as that of a Directive, we cannot solely rely on the results of the NLP processes, which brings us to perform a manual verification in the end.

Automating step 3 We showed two designs in the Structuring step, which set up the structure of the ontology to be developed. In this case, no tools have been employed, since they depend on subjective choices.

However, the first design, which is agent-oriented, can be the result of a preliminary task, which may derive from the Interpretation step: we can appreciate the presence of some clusters when iterating the extraction of the parts of speech throughout the Directive. Such clusters refer to some specific entities, in particular those regarding the subjects of the security measures, namely, *Agent*. Therefore, an automatic tool could take the results of the part-of-speech extraction and, according to some statistical analysis, derive the presence of such clusters. In any case, this would only be useful for a minimal part of the design presented in the first micro-step, since the rest cannot be automatically inferred because the choice depends on the designer of the ontology.

The same considerations may apply to the second micro-step, which does not require particular pre-computation tasks.

Automating step 4 The step consists of two very different micro-steps: the first one introduced to manage the structural insights of a security measure, the second one for the exemplification of the ontological representation. As the first micro-step may directly suggest, this step does not benefit from automatic tools, since similarly to the ones on the previous steps, is based on very subjective decisions that cannot be either selected or supported by automatic tools.
On the contrary, the second micro-step reflects a pure ontological representation, where the main ontological terms arise. For this step, where the security measures are modelled as ontological entities and properties, we adopted the Protégé [126] editor, one of the principal software for ontological modelling.

Automating step 5 For this step, we adopted the Protégé editor. In this particular case, we used three different functionalities: the first one for defining the entities, the second one for the reasoning tasks, and the last one for creating and performing the SPARQL queries. The

inferences obtained are reported in this chapter as the results of each micro-step, but for the sake of presentation, we presented them with a human-understandable syntax.

3.7 Concluding Remarks

This chapter faced the challenge of providing a semantic structure to the newest security directives, further refining the various existing works. The benefits of a semantic representation are considerable, in particular for providing linear structures to such complex documents, and better dissecting the provided security measures and the involved agents.

We have presented `SecOnto`, a thorough methodology designed for representing security directives, particularly those of European origin, into ontologies. `SecOnto` is composed of 5 steps, and each step is composed of several micro-steps, where for each one we provided a brief introduction, a description, and a demonstration on real cases. The `SecOnto`'s steps have been validated with the NIS 2 Directive.

The application of the steps allows an analyst to be driven from the first step of preprocessing of the security document to the last step of leveraging the provided structures for the task of compliance verification. Furthermore, the provided structures and design, thanks to the discussed analysis, could be used in a variety of ways, allowing the resulting ontologies to, e.g., be leveraged to research inconsistencies or the structured research of attacks.

Chapter 4

NIS2Onto: Guiding Security Compliance with NIS 2 Directive

Security is not a product, but a process.

— Bruce Schneier

This chapter is based on the publication *"Guiding Cybersecurity Compliance: An Ontology for the NIS 2 Directive"* [29].

This chapter presents NIS2Onto, the first ontological representation of the full NIS 2 Directive, built with the SecOnto methodology. NIS2Onto is a *Web Ontology Language* (OWL) ontology that can be navigated, inspected and queried by software systems, ultimately guiding the processes of compliance check and research of attacks.

4.1 Introduction

NIS2Onto leverages the intrinsic properties offered by ontologies to arguably reduce the NIS 2 compliance effort, both by enabling continuous monitoring of any modification that may occur during the lifespan of the institution and by countering the risk of error by human personnel. This also implies that fewer resources are required for compliance verification, leading to cost savings and increased efficiency. Moreover, a wide range of compliance requirements can be covered, ensuring that no aspect is overlooked. These requirements may change over time, and such changes are handled automatically without human effort.

The task of representing a Directive through an ontology means, first and foremost, to extrapolate those actions demanded to be adopted to improve the security posture, namely the security measures, and then to deconstruct agents, actions, and objects of such measures. This task may turn out daunting in general and particularly over such a large document written in legal language as the NIS 2, but (it was pleasing to observe that) it became manageable by following SecOnto.

Coherently with what is outlined above, our work is supported by an IT enterprise, Intrapresa S.r.l., [1] operating in the field of innovative payment systems for petrol stations. We acknowledge that one of the case studies to which `NIS2Onto` was applied for the sake of demonstration, discussed below, was provided by this enterprise.

4.2 Related Work

Ontologies are employed mainly in various scientific domains and have been developed substantially. The same stands for cybersecurity — both with a focus on security and legislation — thus we concentrate on the main contributions relevant in both cases, starting with a focus on the security one. Although different from ontologies in the legislative domain, the analysis of purely security ontologies (oriented towards more technical aspects of security, such as systems, networks, protocols, vulnerabilities, etc.) offers an overview of the techniques adopted for their development. Since there are so many related works, we will focus only on those considered most similar, particularly starting from the legislative domain.

To the best of our knowledge, fully representing cybersecurity documents for compliance verification, particularly for the NIS 2 Directive, is an uncharted path. The work that we could consider closest to ours, among all the others, consists of defining some NIS 2 entities, probably the most relevant ones, using the DPV format [59]. Compared to `NIS2Onto`, the work is not about providing a complete representation of the NIS 2 Directive. Considering the possible similarities, it would be reasonable in future research to integrate the two works. Therefore, the effort of `NIS2Onto` aims to exploit a general yet practical and concrete approach to represent and operationalise security directives, ultimately enhancing the effectiveness and efficiency of the compliance process. Drivas et al. [38] presented a framework to measure the maturity level of the requirements against the first version of the NIS directive. The authors do not make use of ontologies, nor do they aim to propose a representation of the entire directive. Instead, they classify under certain categories of controls, the adherence to which determines their maturity level. From the point of view of legal text processing, several works have been proposed regarding GDPR. To provide a legal knowledge modelling of the privacy agents, data categories, kinds of processing activities, rights, and

[1] https://www.intrapresa-it.it/

duties, Palmirani et al. [103] developed PrOnto, a legal ontology on the General Data Protection Regulation (GDPR). This approach is based on ontological patterns combined with an analysis of legal theory. Bartolini et al. [16] have worked extensively on the GDPR encoding, in particular using Reified Input/Output Logic, for example, to be able to correlate ISO/IEC 27018:2014 with GDPR. `NIS2Onto` uses the concept of *agent* to express the entities subject to the security measures defined in the Directive. Although not strictly considered legislative documents, two papers focused on producing ontological representations of three standards, PCI DSS, ISO 27001, and ISO 27002, respectively. In the work of Elluri et al. [41], the regulations required by the Payment Card Industry Data Security Standard (PCI DSS) and EU GDPR have been represented by an integrated, semantically rich knowledge graph. Fenz [47] proposed a methodology for automatically producing IT-security metrics based on ISO 27001, for assessing the efficacy of control implementations and their compliance with information security requirements. Fenz [49] introduced a method for formalising information security control descriptions and a decision support system that enhances automation, thereby improving the cost efficiency of the information security compliance checking process. The author applied this method to ISO 27002 information security controls and created a semantic decision support system. The main differences lie in the fact that `NIS2Onto` takes measurements from a legislative text, which can be quite complex. In contrast, standards by nature are structured in more atomic and self-contained controls. Therefore, although in both cases ontological reasoning is exploited for compliance purposes, NISOnto addresses a different representation problem, closely related to the nature of the legislative measures. In methodological terms, some works have considered the use of external vocabularies for the representation of policies, especially privacy [45]. This is essential when one does not want to provide a slavish representation of the document but rather wants to cluster the concepts and information, e.g., in the GDPR, in entities known in some vocabularies. Always with the idea of managing the difficulties of complex documents, Joshi et al. [68] created a semantically rich ontology and created a database with many policy documents as instances of this ontology. Using deontic logic, the authors identified rules from these policy documents that may be utilised to automate data privacy management. `NIS2Onto` does not aim to create a common vocabulary, nor does it draw on others. The representation of measures requires unique, even non-standard notions that reflect the content of the measures themselves as sentences. Therefore, these may not be reusable in other scenarios, except for possible evolutions of the

directive itself.

Considering the purely security, rather than legislative, domain, we offer a perspective on ontologies that, although not entirely similar to `NIS2Onto`, emphasise how the role of ontologies is employed in the security domain and for different purposes, making ontologies a valuable tool. Tailhardat et al. [131] developed the NORIA-O ontology to define an infrastructure, its events, diagnostics, and remediation operations carried out during incident management. An example use case detailing a hypothetical failure illustrates how this ontology may be used to describe intricate scenarios and provide a foundation for anomaly identification and root cause investigation. Syed et al. [130] presented the UCO ontology. UCO is designed to help cybersecurity systems with information integration and cyber situational awareness. To facilitate information sharing and exchange, the ontology combines and integrates diverse data and knowledge schemas from many cybersecurity systems with the most widely used cybersecurity standards. Muñoz et al. [96] developed the PPROC ontology, which is designed to provide semantic descriptions of public procurement contracts and procedures, supporting disclosure and accountability. The PPROC ontology is comprehensive as it includes information on the whole process, from the first contract publication to its termination, in addition to the standard information on the tender, its goals, deadlines, and recipients. Syed [129] presented a conceptual model for formal knowledge representation of the vulnerability management area, which is the Cybersecurity Vulnerability Ontology (CVO). CVO is used to create a Cyber Intelligence Alert (CIA) system that sends out cyber warnings on vulnerabilities and countermeasures. Wang et al. [137] developed OVM, which is composed of the vulnerabilities in the National Vulnerability Database (NVD), for vulnerability management (OVM), along with extra inference rules, knowledge representation, and data-mining techniques. OVM offers a potential road to the success of the Information Security Automation Program (ISAP) through the smooth integration of common vulnerabilities and their associated concepts, such as attacks and solutions.

Table 4.1 provides a summary of the most prominent works in the field, emphasising whether each ontology pertains to a pure security domain or a legislative context.

Table 4.1: Summary of related work

Citation	Title / Ontology	Security	Legislative	Description
Fenz (2018) [49]	Semantic DSS for ISO 27002	✓	✗	Automates compliance checks using security control descriptions.
Tailhardat et al. (2023) [131]	NORIA-O	✓	✗	Incident management, diagnostics, and remediation modeling.
Syed et al. (2016) [130]	UCO	✓	✗	Facilitates cyber situational awareness and data integration.
Muñoz et al. [96]	PPROC	✗	✓	Focus on public procurement contracts and procedures.
Syed (2020) [129]	CVO	✓	✗	Supports cyber warning systems for vulnerabilities.
Wang et al. (2009) [137]	OVM	✓	✗	Ontology for vulnerability analysis based on NVD.
Palmirani et al. (2018) [103]	PrOnto	✗	✓	Legal modeling of GDPR concepts using ontologies.
Elluri et al. [41]	GDPR + PCI DSS Knowledge Graph	✓	✓	Integrates GDPR/PCI DSS with CSA security controls.
Pandit et al. (2022) [45]	External privacy vocabularies	✗	✓	Clusters privacy-related concepts using vocabularies.
Joshi et al. (2016) [68]	Privacy policy ontology	✓	✓	Extracts rules from privacy documents using deontic logic.
Matsunaga et al. (2017) [88]	Data anonymisation ontology	✓	✓	Standardises anonymisation policy for big data/cloud.
Castiglione et al. (2025) [27]	SecOnto / `NIS2Onto`	✓	✓	Methodology for converting NIS 2 into operational ontologies.
W3C DPV [59]	NIS 2 partial in DPV	✗	✓	Only models some entities from the directive using DPV.

Since NIS2Onto is specific to the domain of security directives of the NIS 2 and tailored to the sections where the security measures are expressed, no foundational ontology has been initially considered for the development of NIS2Onto, even though some are undoubtedly useful for integrating the work with other contributions. Additionally, most of them are mainly devised for contexts outside cybersecurity, while, to the best of our knowledge, there are no significant contributions aimed at representing security documents. Some ontological efforts rely on technological systems and processes, but these cannot be considered for the current stage of NIS2Onto development. This does not exclude that upcoming versions of NIS2Onto may adapt to domain-related ontologies, even in correspondence with the evolution of the NIS 2 Directive itself.

4.3 The NIS2Onto Ontology

4.3.1 NIS2Onto overview

The construction of NIS2Onto is entirely based on all the steps foreseen by SecOnto: the first 4 steps for choosing the security measures to be considered and their transformation into an ontological format, the last step for the exploitation of the representation obtained for compliance verification activities — a case study of which is presented in this chapter.

By applying the *Preprocessing* step of SecOnto, we identify the Chapters from II to VII, composed of the Articles by 7 to 37, as the most suitable for the ontological representation.

By applying the *Interpretation* step, NIS2Onto identify the agents responsible for the security measure (who must carry out the action described in the security measure determined by the subject), the action to be carried out to comply with the security measure (the verb of the security measure), and the object or series of objects of the security measure, namely, the recipient of the security measure in terms of either further agents or further processes/elements in general, identified by the object of the security measure. An agent is any subject to which any NIS measure refers; therefore, an agent is either an institution (Member States, European security organisations) or an organisation/company. They are considered agents because they must act in organisational and strategic terms to fulfil the NIS measures prescribed to them.

By applying the *Structuring* step, NIS2Onto reflects the principle of SecOnto that associates specific agents with the security measures the agents must fulfil through the equivalence relationship, i.e., *EquivalentTo*,

with a class description devising the features of the security measures, the object properties representing the actions and entities representing the object of the actions. Since NIS2Onto reflects the document-oriented design from SecOnto, this step is done for each security measure. Within this structuring, we can refer to both the security measure of the document, the related Articles, Chapters and Paragraphs, and to the agent and the related actions and recipients. This is particularly useful for keeping the ontology aligned with the document, identifying the measure's original location and description and enabling a semantic search of its constitutional elements, as Figure 4.1 and Figure 4.2 partially illustrate.

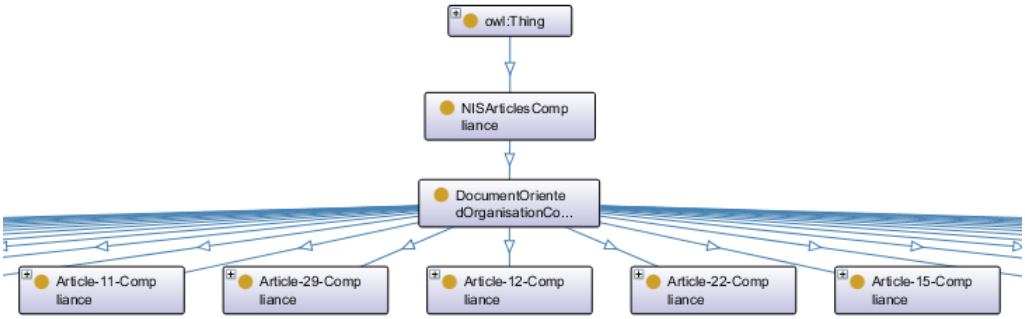


Figure 4.1: Excerpt of Articles in the Document-oriented Representation

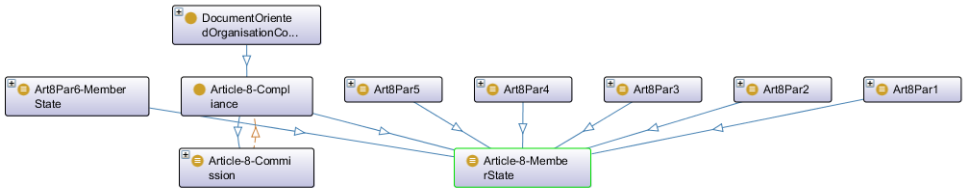


Figure 4.2: Excerpt of Articles and Paragraphs in the Document-oriented Representation

By applying the *Representation* step, NIS2Onto handles the very complex structures of the security measures, due to the legal nature of the Directive. This particularly applies when associated with the main agent of a measure; there are, e.g. multiple objects, leading to multiple levels of characterisation of the actions themselves. One of the possible cases

identified by SecOnto in the current step is illustrated in the Figure 4.3, where a multiplicity of objects from the hierarchical structure of the same measure (Article 7 Paragraph 2), associated with the same agent (MemberState), is represented.

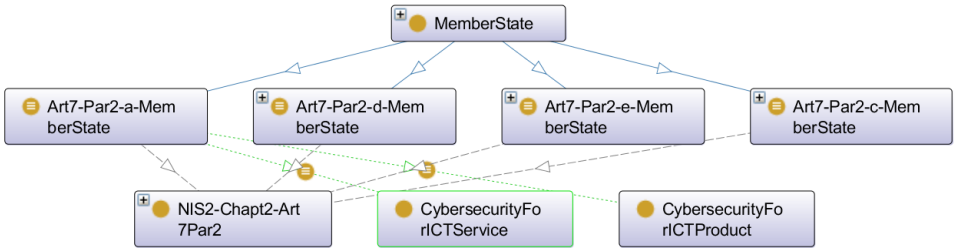


Figure 4.3: Excerpt of Articles and Paragraphs in the Document-oriented Representation

4.3.2 Classes and individuals

As stated before, the goal of NIS2Onto is to represent the agents and objects of the security measures, keeping intact the structure of the document. For this purpose in NIS2Onto, there are four main conceptual families of classes:

Document These classes are useful for coherently maintaining the document structure and eventually, for associating the articles and *Paragraphs* with the related agent. *Chapters* are organised in *Articles*, Articles in Paragraphs, where each Paragraph refers to specific agents, actions and objects. The structural organisation of the document is reflected in a suitable hierarchy of classes, thus allowing the association of agents to the security measures in the document where they are mentioned. This association is carried out by means of the *referTo* object property.

One of the document-related classes is *CybersecurityDirective*, which contains a list of European directives in Cybersecurity, including the current NIS 2, the *NIS2-Directive* class. This class contains the classes representing the NIS2 chapters (e.g., *Chapt1, Chapt2, etc.*). In their turn, chapters are organized in articles (e.g., Art7, Art8, etc.) and, eventually, in Paragraphs (e.g., Art7Par1, Art7Par2, etc.). Each Paragraph may be in turn specialised in items, if a list exists (e.g., Art7Par1-1, Art7Par1-b, etc.)

Agent For the NIS 2 Directive, it is crucial to identify the agents involved in the provided security measures. NIS2Onto has been

developed by focusing on this goal, in particular, on the agents relevant to the NIS 2 Directive, for which compliance should be verified. Therefore, the ontology provides suitable classes, one for each type of agent, where the related instances model the agents depicted in the Directive. Some agents are introduced outside the NIS Directive (e.g., Member State), while others are introduced specifically within the Directive (e.g., Eu-Cyclone).

Object The family of objects consists of those elements that uniquely represent what the objective of a security measure is, i.e. the object of the measure itself. These classes reflect what the action of an agent (represented by the verb of a security measure) should be. As in the case of the agents, the objects of the security measures are introduced by way of a suitable hierarchy. In most cases, it is hard to identify the object of a security measure and to associate it with a specific category. The class names adopted have been obtained by removing the articles, adverbs, and other grammatical structures from the object name.

Considering two cases, security measures may define objects that are either easily extractable or more complex, depending on the nature of the measure itself and the trade-off in naming the class to ensure sufficient clarity. The class *NationalCybersecurityStrategy* (the measure states that the *Member State shall adopt a National Cybersecurity Strategy)*, representing the object of the first measure of Article 7, is a clear example of a class that falls into the first category. In contrast, the class *ThirdCountriesNationalComputerSecurityIncidentResponseTeam* (the measure states that *the CSIRTs may establish cooperation relationships with third countries' national computer security incident response teams*), from paragraph 7 of Article 10, represents a more complex object, requiring some text processing to fully grasp its meaning.

Compliance `NIS2Onto` classifies agents depending on the measures they are compliant with. The classes of the compliance family serve to preserve the relationships between the agents and the measures to which they must respond, thus being compliant. These classes serve to reflect both the name of the agent to whom the measures are addressed and the measures themselves, precisely identified by the specific article and paragraph in which they are contained. For instance, CSIRT agents compliant with Article 10, Paragraph 4 of the NIS 2 Directive are defined as instances of a class named

Art10Par4-CSIRT. Their membership in *Art10Par4-CSIRT* is inferred through suitable class expressions derived from the Directive. The same holds for other agents (MemberState, ENISA, Eu-Cyclone, etc.) to whom the security measures are addressed.

Each class is suitably defined to check for the measures that must be satisfied by the compliant entity. For example, let us consider an excerpt of Article 10 from the NIS 2:

> **Computer security incident response teams (CSIRTs)**
>
> 1. Each Member State shall designate or establish one or more CSIRTs. The CSIRTs may be designated or established within a competent authority. The CSIRTs shall comply with the requirements set out in Article 11(1), shall cover at least the sectors, subsectors and types of entity referred to in Annexes I and II, and shall be responsible for incident handling following a well-defined process.
>
> 2. Member States shall ensure that each CSIRT has adequate resources to carry out effectively its tasks as set out in Article 11(3).
>
> 3. Member States shall ensure that each CSIRT has at its disposal an appropriate, secure, and resilient communication and information infrastructure through which to exchange information with essential and important entities and other relevant stakeholders. To that end, Member States shall ensure that each CSIRT contributes to the deployment of secure information-sharing tools
>
> ...

To verify the compliance of a Member State with Article 10, the Member State should satisfy each of its Paragraphs. A Member State, to be compliant with Article 10, should adopt, i.e., be compliant with, the Paragraphs defined in the Article itself. For this reason, in `NIS2Onto` exists a class named *Article-10-CSIRT* which is equivalent to the classes representing the paragraphs, as the following measure represents.

```
Art10Par1-CSIRT and Art10Par10-MemberState
 and Art10Par2-MemberState and Art10Par3-MemberState and
 Art10Par4-CSIRT and Art10Par5-CSIRT and Art10Par6-MemberState
 and Art10Par7-MemberState  and Art10Par8-CSIRT
 and Art10Par9-MemberState
```

The definitions imply that if and only if the Member State adopts *all* the measures defined in the Paragraphs of Article 10, then it results in being *Compliant* with *Article 10*. This is a clear example of how the word *Compliant* may result in more significance than the word *Compliance*.

If we inspect the class representing Paragraph 3, namely *Art10Par3-MemberState*, we notice that complying with it derives from satisfying the security measures described in it, as the following shows:

```
MemberState
 and ((allowTo some EntityAndStakeholderInformationExchange)
 and (ensureCSIRTContributesTo some InformationSharingToolsDeployment)
 and (ensureCSIRTHas some CommunicationInformationInfrastructure))
```

The same considerations apply to the other Paragraphs, and consequently, Articles of the NIS 2. In these cases, *MemberState* and *CSIRT* appear in the class name to highlight the importance of the involved agent.

4.3.3 Object-properties

In `NIS2Onto`, the object properties are used to identify the actions taken by an agent concerning some object. In the context of a security measure, the object property is introduced by the main verb of the measure, namely the verb strictly related to the agent of the selected measure. Since there are different types of actions, we adopt the following naming convention.

- The object property name reports the verb itself. This case occurs when the object is well-established, and it is not necessary to mediate by other terms.
- The object property name is obtained by the verb, but it includes other additional elements. This occurs when the action is complex or the verb is phrasal. For example, *allow* falls in the first case, while *allowTo*, *allowToAccess*, *allowToEntitiesTimeTo* fall in the second one. As their names may suggest, these more complex object properties serve to lighten and convey the writing of the object as best as possible, which, thanks to this approach, may remain devoid of details that could only be useful for the flow of the measurement but which would overload the meaning of the object itself. Although this may reduce the re-usage of the objectives for other security-related vulnerabilities, we notice that in *almost* all object properties cases, the

most complex ones are always based on already existing simple objects. For example, whether the object *allowToEntitiesTimeTo* may seem non-reusable, it is based on the simpler property *allow*, which can be extended to external developments.

4.3.4 Data-properties

In the NIS 2 Directive, numerical values can be expressed as objects of some actions, e.g., for establishing data breach notification months, or the period for submitting the national cybersecurity strategy. In such cases, data properties are adopted to express the actions with the same name convention adopted for the object properties.

4.3.5 SWRL rules

SWRL (Semantic Web Rule Language) is a rule-based language that extends the expressive capabilities of OWL ontologies by allowing the specification of Horn-like rules. These rules enable more complex inferences that go beyond what OWL alone can express, such as property chaining, conditional logic, or constraints across multiple individuals. SWRL operates directly over OWL classes, properties, and individuals, which means it integrates seamlessly with existing semantic models without altering the OWL syntax or semantics. Rather than introducing a new reasoning paradigm, SWRL complements OWL by enabling rule-based reasoning within the same framework. We specifically identify the following two scenarios which `NIS2Onto` can benefit from the introduction of SWRL rules:

Self-referencing entities The first scenario regards the presence of self-referencing entities inside a security measure. For instance, the following rule

$$\texttt{MemberState}(?\,\texttt{x}) \wedge \texttt{CompetentAuthority}(?\,\texttt{y}) \wedge \texttt{designate}(?\,\texttt{x}; ?\,\texttt{y}) \wedge \texttt{PointOfContact}(?\,\texttt{z}) \rightarrow \texttt{isSinglePointOfContact}(?\,\texttt{z}; ?\,\texttt{x})$$

ensures that a security measure must be linked with the specific *Member State* to which it pertains. This approach becomes valuable during the compliance verification step. For instance, if multiple *Member States* are instantiated and associated within the related security measure, the rules will trigger the inconsistency. Essentially,

this rule reinforces the connection between the measure and the specified *Member State*.

Describing preconditions and implications The second scenario concerns specific measures of the Directive that might encompass structures that involve not only explicit security measures but also conditional elements that prescribe the application of these measures. These conditional clauses introduce intricacies into both the interpretation of the Directive and the representation of such measures. As an example, we consider the following measure (Article 26, third Paragraph, first sentence):

> "If an entity as referred to in Paragraph 1, point (b), is not established in the Union but offers services within the Union, it shall designate a representative in the Union."

The measure is guaranteed by the following rule:

$$\begin{aligned} &\texttt{Entity}(?\,\texttt{x}) \wedge \texttt{NotEuropeanUnion}(?\,\texttt{x}) \wedge \texttt{referredTo}(?\,\texttt{x};?\,\texttt{y}) \\ &\quad \wedge \texttt{offerServicesTo}(?\,\texttt{t};?\,\texttt{x}) \wedge \texttt{UnionRepresentative}(?\,\texttt{u}) \\ &\qquad \wedge \texttt{EuropeanUnion}(?\,\texttt{t}) \wedge \texttt{Art23Par3-b}(?\,\texttt{y}) \\ &\qquad\quad \rightarrow \texttt{designate}(?\,\texttt{u};?\,\texttt{x}) \end{aligned}$$

4.3.6 Industrial Exploitation of NIS2Onto

The provided representation of the security measures offers a dual standpoint for the critical and crucial task of compliance verification. Institutions and organisations can consider the ontology as a basis for further developments closely related to their cases and implementations of both security and organisational measures. This is possible thanks to the thoroughness of the domain, allowing for the use of both a top-down and bottom-up approach in evaluating individual instances. The first approach is based on the created ontology without any modification of its classes or properties and essentially follows exactly what is shown in the methodology. It consists of creating the individuals to be verified, assigning the security measures to them, again using the instantiating of classes and properties, and then evaluating the inferences generated by the reasoning. Conversely, the second approach takes a divergent path by leveraging and expanding upon the NIS 2 notions of *sector* and *entity*, which hold pivotal significance in achieving the objectives outlined in NIS 2. Notably, NIS 2 is explicitly geared towards addressing entities

considered as crucial components within a nation's critical sectors. This approach delves deeper into ontology development by introducing new classes encapsulating the most pertinent existing standards. This distinctive approach is tailored to foster a context-specific evaluation of the business landscape, with a primary focus on the standards currently in practice. It is essential to note that the NIS 2 Directive refrains from imposing the adoption of specific standards, providing each enterprise with the autonomy to make its own choices. Consequently, referencing the measures of a Directive can initially appear abstract in the absence of classes that represent concrete standards. The fundamental premise here is that when a specific standard is adopted, adherence to the security measures associated with that standard is inherently guaranteed, thus substantiating compliance with those measures. Building upon this concept, the idea materialises in creating these missing classes. For instance, one might have a subclass like *PCI-DSS* for class *Multi-factor Authentication* or subclass *NIST SP 800-53* for class *CryptographyPolicy*. With the introduction of such classes, the compliance verification task is twofold. Initially, it involves a more intricate internal audit where the specific standards in use are systematically categorised. Subsequently, harnessing the power of inferences, these specialised classes become directly associated with the broader concept expressed within a security measure. This approach, which aligns compliance with specific, adopted standards, further refines the verification step and empowers a more characteristic understanding of adherence to NIS 2 measures.

4.3.7 Ontology Maintenance and Usability

Maintenance The maintenance activity of `NIS2Onto` essentially foresees the manual scrutiny of experts. The maintenance activity would not be particularly burdensome because the directives are more static objects, which change infrequently over time. However, even version changes, e.g. the transition from the first version of NIS [135] to the second, i.e. NIS 2 [43], brought about changes, but not so radical as to modify the structure of the new NIS 2 Directive. The literature proposes different perspectives on the maintenance of ontologies, each dictated by the nature and structure of the ontologies themselves [109]. Therefore, the maintenance activity would either eliminate and/or add new security measures as atomic operations on the ontology [52], or refine the current measures towards a different interoperability with those who would employ `NIS2Onto`, as ontology evolution [52]. This process would not be particularly suitable to be delegated to automated tools, due to the high

level of detail present.

Usability NIS2Onto is a representation of the NIS 2 Directive designed to exploit the logical and representational properties of ontologies to provide an accurate and, above all, interoperable version of the Directive. The present work does not focus on how NIS2Onto can be used with user-friendly applications, but demonstrates how the current version, although technical and challenging for non-experts in the field, is workable thanks to the underlying capabilities of reasoning. Starting from this already functioning basis, various uses of NIS2Onto can therefore be thought of. Two possible directions can be identified: using NIS2Onto as a structured source for the integration with LLMs; using NIS2Onto as a backend part of compliance services. While the second case would not bring effective contributions to research, the first case is especially compelling, since research activities are focusing on Neurosymbolic AI [121]. An ontological structuring, such as that of NIS2Onto, would be particularly relevant, by providing an accurate and explainable [111] knowledge layer for enhancing the interpretability and reliability of LLMs. This is particularly relevant in the legal domain; although the LLMs have increasingly higher capacities, they may be limited in correctly interpreting documents written in legal language, with structural and semantic complexities, of which NIS 2 is an example.

4.3.8 NIS2Onto Queries

On top of NIS2Onto [56], the following competency statements may be translated into queries accordingly. In Appendix A, we present the queries for the categories *Differential Analysis*, *Specific Search*, and *Integration*, while the query for *Compliance Check* is illustrated together with the case study.

1. **Compliance check I**. We want to check the compliance of a certain individual with the Directive.

2. **Compliance check II** We may want to check the compliance of an individual concerning a specific article.

3. **Differential analysis I**. We may want to verify or search for the security measures that involve some entity (undertaking vs. entity).

4. **Differential analysis II**. We may want to verify or search for the agents that are involved in some security measures (undertaking vs. important or essential).

5. **Specific search I**. Given an object (e.g., the risk assessment), we may want to search for the article that treats it.

6. **Specific search II**. Given an object (e.g., the risk assessment), we may want to search for the action that involves it.

7. **Specific search III**. We may want to search for all objects syntactically containing a certain word to verify which elements are involved.

8. **Specific search IV**. We may want to search for the standard associated with a specific category (e.g., vulnerability assessment, risk assessment, inconsistencies management).

9. **Integration I**. We may want to integrate the measures of common agents among other security directives/regulations, e.g., GDPR.

10. **Integration and Differential Analysis**. We may want to check the security measures missed by an agent across several directives/regulations.

4.4 `NIS2Onto` Evaluation

While we have experimented with more widely known evaluation tools on our ontology, the results proved to be unreliable. This is because these tools [139] are structurally incompatible with the specific nature of `NIS2Onto`, which, as mentioned above, is a domain-related ontology focusing on the representation of the NIS 2 Directive. On the contrary, a foundational ontology should be applicable across many domains and provide very abstract categories that can be specialised by domain ontologies.

Nevertheless, though not relying on a foundational ontology, our model is built upon a well-defined, recursive, and detailed logical structure. Therefore, we have chosen to focus on both a quantitative and qualitative evaluation, grounded in the ontology itself.

Metrics serve as guidelines for evaluating the complexity, quality, and usability of ontologies, facilitating their development and refinement, and additionally offering insights into characteristics such as the number

of classes, properties, instances, and logical axioms. Ontological metrics enable practitioners to gauge the structural richness and coherence of an ontology. This not only aids in the comparison and selection of ontologies for specific applications but also supports the iterative improvement of ontological models, ensuring they effectively represent the intended domain knowledge. On the current version of the ontology, the following metrics derived from OntoMetrics [90] have been computed:

Base Metrics

- Axioms: 3632
- Logical axioms count: 1798
- Class count: 1330
- Total classes count: 1330
- Object property count: 363
- Total object properties count: 363
- Data property count: 24
- Total data properties count: 24
- Properties count: 387
- Individual count: 44
- Total individuals count: 44
- DL expressivity: ALCHQ(D)

Class Axioms

- SubClassOf axioms count: 1339
- Equivalent classes axioms count: 358
- Disjoint classes axioms count: 2
- GCICount: 0
- HiddenGCICount: 352

Data Properties Axioms

- SubDataPropertyOf axioms count: 10
- Data property domain axioms count: 1

Individual Axioms

- Class assertion axioms count: 43
- Object property assertion axioms count: 45

Annotation Axioms

- Annotation assertion axioms count: 72

Schema Metrics

- Attribute richness: 0.018045
- Inheritance richness: 0.962406
- Relationship richness: 0.361277
- Equivalence ratio: 0.269925
- Axiom/class ratio: 2.687218
- Class/relation ratio: 0.663673

Knowledge-base Metrics

- Average population: 0.033083
- Class richness: 0.030827

We can interpret the values obtained as follows:

- **Logical Axioms/Total Axioms**. A ratio of logical axioms to total axioms close to 49.5% suggests a fair level of formal logic without being overly axiomatic. This indicates that `NIS2Onto` maintains a balanced design.
- **Attribute richness**. An attribute richness value of 0.018 indicates few data-type properties per class. This appears reasonable given the NIS 2 domain, which emphasises relationships and categorical structures over literal attributes.
- **Inheritance Richness**. An inheritance richness value of 0.962 suggests that most classes participate in subclass hierarchies, indicating that the ontology exhibits strong taxonomic depth.
- **Relationship Richness**. A relationship richness value of 0.361 can be considered moderate, as it indicates that approximately 36% of the schema axioms are object properties. This value is likely since classes describing compliance measures are well defined, whereas those representing subjects and objects are not. As future work, we plan to investigate how to formally describe these latter classes.
- **Equivalence ratio**. An equivalence ratio of 0.66 indicates a moderate use of the *owl:equivalentClass* construct.
- **Class/relation ratio**. A class-to-relation ratio of 0.66 is generally acceptable. A high value may indicate underspecified relationships, whereas a low value could suggest an insufficient number of classes.

Moreover, the presence of 1,339 *SubClassOf* axioms closely matches the total number of classes, while 358 *EquivalentClass* axioms and 352 hidden GCIs further attest to `NIS2Onto`'s well-balanced and semantically rich structure. Conversely, the average population value of 0.003 and the class richness of 0.0031 are relatively low. These figures are consistent with the ontology's intended purpose, as `NIS2Onto` is primarily designed as a schema to be instantiated with user-provided data for compliance verification, rather than to contain extensive predefined individuals or literal attributes.

As stated above, the evaluation of the ontology focuses on qualitative rather than quantitative aspects since the metrics associated with the ontology could be quite variable in correspondence with its possible modifications and changes. Although such assumptions may not seem

entirely accurate in an ontological context, their justification derives from the nature of NIS2Onto, namely, a domain-related ontology that may not be fixed over time. For example, if a company wanted to use NIS2Onto to verify compliance with its requirements and wanted to break down an object of a security measure into smaller objects, then it would be able to do so, without destroying the meaning of the structures provided. It is therefore clear that the metrics would no longer be consistent if such a scenario occurred. Instead, the evaluation of the current version of NIS2Onto follows some best-known approaches classified by Raad et al. [112]:

- **Corpus-based**: *"Corpus-based approaches are used to evaluate how far an ontology sufficiently covers a given domain"*. NIS2Onto is an application-based ontology. It represents a specific context as it is created ad hoc on the NIS 2 by covering the articles from 7 to 37, the articles presenting security measures. Hence, by coverage all the meaningful 31 articles, the NIS 2 is entirely covered.
- **Task-based**: *"Task-based approaches try to measure how far an ontology helps to improve the results of a certain task"*. The main aim of this chapter is to provide a methodology for a representation that can be easily intelligible, make the entities and properties reusable with the integration of further ontologies, and provide mathematical support for compliance verification. Although there may be tools that assist with compliance verification, they may not accomplish the peculiarities on which the ontologies are based, and which have already been contextualised.
- **Criteria-based**: *"Criteria-based approaches measure how far an ontology or taxonomy adheres to certain desirable criteria"*. The methodology is particularly suitable for obtaining structures that provide a representation that is accurate, concise, complete, efficient, and more generally respectful of the FAIR principles.

Concerning task-based approaches, for the NIS 2 Directive, we currently focus on *(a) Question Answering*, *(b) Reasoning for decision support*, and *(c) Information Retrieval*, whereas *(d) Information Retrieval*, *(e) Data Integration*, and *(f) Interoperability between systems* lie outside the present scope of this work, as stated above, since they require a comprehensive framework to be carried out. Our main interest lies in how companies can verify their alignment with NIS 2 and, if they are not compliant, how they can identify the missing security measures.

For this purpose, we design the case study presented in Section 6. Concerning criteria-based approaches, we recall the following dimensions: *(a) Accuracy* – the alignment of concepts and relationships with the domain; *(b) Completeness* – the extent to which the ontology covers the domain; *(c) Conciseness* – the absence of redundancy; *(d) Adaptability* – the modularity and flexibility of the ontology; *(e) Clarity* – the unambiguity of labels; and *(f) Efficiency* – the performance of reasoning tasks. Accuracy, completeness, conciseness, and clarity were ensured manually, as the ontology was derived directly from the NIS 2 directive. The **adaptability** of the ontology stems from its ability to accommodate alignment with future versions of the *NIS 2* directive; while the adaptation of the ontology currently relies on expert intervention, some tools (see above) can be further investigated to increase the level of automation. Finally, **efficiency** is supported by a case study derived from a real-world context.

4.5 Case Study

This section shows how to apply `NIS2Onto` to a real case study developed in the context of the business activities of Intrapresa S.r.l. In the considered case, the company would check its compliance with the NIS 2 Directive. Of course, this Section does not show the actual company's compliance obligations to preserve its confidential business information — the data that is shown is appropriately anonymised, namely randomised by permuting the values of the obligations.

The infographics in Figure 4.4 portray a company that integrates `NIS2Onto` in its compliance verification framework. It can be seen that first the company populates the ontology with relevant company information, focusing on the security measures already fulfilled. To do this, the company maps the obligation in the corresponding OWL assertion involving it. The reasoner is then employed to infer and complete missing knowledge. Subsequently, a dedicated SPARQL query is used to identify outstanding security measures. This process ultimately enables the company to receive a clear roadmap for achieving full NIS 2 compliance. These steps can be carried out either interactively or programmatically. To present the results, we adopt the Protégé [95] ontology editor, but any compatible tool can be used.

By following the ontology specification, we create a fresh individual that represents the company. As a first step, we first introduce the company as an instance of *Entity*, and then we describe all the features representing the company, namely all the security measures the company

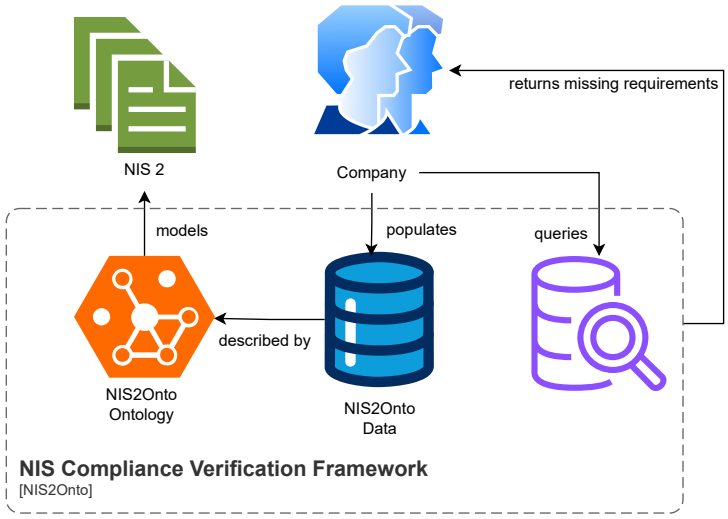


Figure 4.4: The compliance workflow of a company that adopts `NIS2Onto`

currently satisfies. These are depicted in Figure 4.5. For example, since the company applies some strategies of disaster recovery, we state that it "include DisasterRecovery".

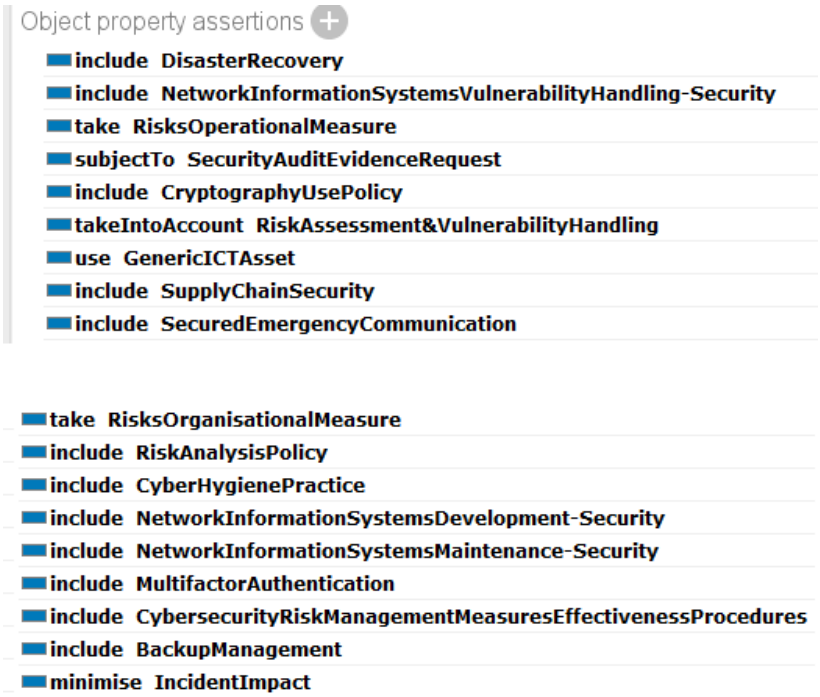


Figure 4.5: Set of security measures satisfied by the company

By exploiting the inferences from the reasoning task on the ontology, we verify which Articles the company complies with.

Hence, we exploit the reasoning capabilities of the ontology to find out which Articles the company is compliant with, and we can leverage them to understand which security measures are not satisfied, if any. In this case, the ontological entity representing the company is inferred as an instance of the compliant article; otherwise, such an inference is not entailed. In our case study, the company is currently satisfying the compliance measures partially shown in Figure 4.6, which are obtained thanks to the ELK reasoner [71]. [2]

The security measures satisfied by the company are introduced through the mapping in Table 4.2. The first column represents the verbal description of the satisfied obligation, while the second column shows the corresponding OWL assertion involving the created entity.

[2]ELK is an OWL 2 EL-compliant reasoner that offers the best trade-off between expressivity and performance, given the large number of OWL restrictions defined in `NIS2Onto`.

Table 4.2: Ontology mapping of the company's adopted security measures

Security and Risk Management Strategy	OWL Assertion
Adopts disaster recovery strategies	include DisasterRecovery
Adopts strategies for handling network vulnerabilities	include NetworkInformationSystems VulnerabilityHandling-Security
Adopts operational risk measures	take RisksOperationalMeasure
Adopts security audit measures	subjectTo SecurityAuditEvidenceRequest
Adopts cryptographic techniques	include CryptographyUsePolicy
Adopts risk assessment strategies	takeIntoAccount RiskAssessment&VulnerabilityHandling
Adopts ICT asset management practices	use GenericICTAsset
Adopts supply chain security measures	include SupplyChainSecurity
Adopts security emergency communication procedures	include SecuredEmergencyCommunication
Adopts organizational risk measures	adopt RiskOrganisationalMeasure
Adopts risk analysis policies	take RiskAnalysisPolicy
Adopts cyber hygiene practices	include CyberHygienePractice
Adopts secure network development measures	include NetworkInformationSystems Development-Security
Adopts secure network maintenance measures	include NetworkInformationSystems Maintenance-Security
Includes multi-factor authentication mechanisms	include MultifactorAuthentication
Adopts cybersecurity risk management procedures	include CyberSecurityRiskManagementMeasuresEffectivenessProcedures
Adopts backup management practices	include BackupManagement
Seeks to minimize incident impact	minimise IncidentImpact

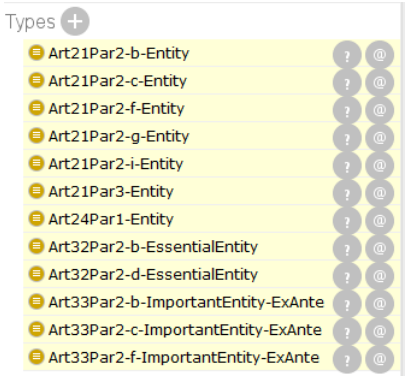


Figure 4.6: Compliance measures satisfied by the company.

When the company satisfies specific measures, the corresponding individual can be inferred as the NIS 2 agent prescribed to respect those measures. In our case study, the reasoner infers that the company is an

instance of *ImportantEntity-ExAnte*, since the individual is compliant with all the measures prescribed to a NIS 2 Important Entity; additionally, it is also *Ex-ante*, a term used to introduce those entities whose security measures are defined before a security incident occurs.

If there are some missing features, the reasoning shows only the measures the entity complies with, as in Figure 4.6. Afterwards, we can exploit those measures to conduct a differential analysis to understand which Articles the company is not compliant with. In our case, we perform the query corresponding to competency question 2 of Section 4.3 to discover which measures the company is not compliant with. We can also find which actions should be taken by the company to fulfil such measures and therefore be fully compliant with the NIS 2 Directive, thanks to purpose-built SPARQL queries. For example, the SPARQL query in Query B.3 (already illustrated within `SecOnto`, paragraph 3.4.5.3) is introduced to compute the requirements that the company is missing to be compliant with the NIS 2. Specifically, it computes a set subtraction between the measures specified by the compliance model, in our case, *ImportantEntity-ExAnte*, and the measures to which the provided entity is compliant. As shown in Figure 4.7, these are Article 21, Paragraphs 1 and 2, and Article 33, Paragraph 2 (first column). The query also shows the actions to be taken (second column) and the objects to which the action should be applied (third column) to attain full compliance: this interaction increases the company's awareness of the tasks to carry out towards compliance.

Code 4.1: SPARQL query for compliance check

```
SELECT ?article ?action ?object
WHERE {
  nis:ImportantEntity-ExAnte owl:equivalentClass ?a .
  ?a owl:intersectionOf ?b .
  ?b rdf:rest* ?c .
  ?c rdf:first ?article .
  ?article owl:equivalentClass ?e .
  ?e owl:intersectionOf ?f .
  ?f rdf:rest* ?t .
  ?t rdf:first ?s .
  ?s owl:onProperty ?action .
  ?s owl:someValuesFrom ?object .
  MINUS {
    nis:CompliantOrganisation ?action ?objInd .
    ?objInd rdf:type ?object .
  }
}
```

article	action	object
Art21Par1-Entity	minimiseOn	ServicesRecipientIncidentImpact
Art21Par2-a-Entity	include	InformationSystemSecurityPolicy
Art21Par2-d-Entity	include	EntityProviderRelationship-SecurityAspect
Art21Par2-e-Entity	include	NetworkInformationSystemsAcquisition-Security
Art21Par2-h-Entity	include	CryptographyUseProcedures
Art21Par2-j-Entity	include	SecuredVideo
Art21Par2-j-Entity	include	SecuredVoice
Art33Par2-e-ImportantEntity-ExAnte	subjectToRequestFor	InformationAccess

Figure 4.7: Result of the SPARQL Query

4.6 Automating NIS2Onto

The automation of NIS2Onto is essentially related to two different parts: creation of the ontology and compliance verification. For the creation of the ontology, we should have an efficient and performant NLP pipeline capable of correctly extracting the Part of Speech from the Directive, as in our case. However, we have already shown within GTCheck how the state-of-the-art NLP tools are not fully capable of correctly accomplishing this task.

Instead, compliance verification is automatic at its core, since the compliance verification employs logical reasoning to derive the overall compliance (and after the missed security measures, when applied, the queries) from a company's point of view. Naturally, a security operator should perform manual auditing to "tick" the addressed measures. However, we can consider this a company achievement rather than a methodological one, and how capable the company is in automating and aggregating the security information that flows within the company itself.

4.7 Concluding Remarks

This chapter presented NIS2Onto, an OWL ontology designed to model the NIS 2 Directive to guide the cybersecurity compliance process effectively and structured research of attacks. Our work addresses the pressing need for a structured and interoperable framework that can effectively model legal and technical requirements, facilitating compliance and enhancing understanding across diverse stakeholders. NIS2Onto provides a comprehensive representation of the key concepts of the Directive, including entities, relationships, and security measures, thereby offering a robust tool for compliance management. NIS2Onto enables

the automation of compliance verification processes, supports risk assessment, and fosters improved communication among cybersecurity professionals, legal experts, and organisational leaders.

The evaluation of NIS2Onto, using a combination of ontological metrics and qualitative analysis, demonstrates its effectiveness in encapsulating the requirements of the Directive and its potential for practical application. While ontological metrics offer insights into the structural attributes of NIS2Onto, the contribution of ontology is to facilitate nuanced interpretations of complex legal texts and support decision-making processes. As a result, it seems fair to claim that NIS2Onto has reached maturity in the region of TRL 4-5, which implies that more thorough multi-user and multi-scenario testing remains outstanding.

Chapter 5

WISARD: Semantic Methods for Deriving Attack Patterns from Security Directives

Any sufficiently advanced technology is indistinguishable from magic.

— Arthur C. Clarke

This chapter illustrates WISARD, a methodology for deriving CAPEC attack patterns leveraging the non-compliance with the security measures from the NIS 2 Directive. WISARD employs two *semantic methods*, i.e. two methods that make use of the concept of semantics with different connotations; respectively, semantic similarity and ontological semantics. The two semantic methods aim to derive attack patterns from security measures that are *semantically* correlated. The obtained correlations by the semantic methods are cross-referenced with a validation base obtained with the involvement of security experts.

5.1 Introduction

Security risks arising from the failure to adhere to established directives must be carefully considered, as such non-compliance can reasonably be assumed to increase the likelihood of attacks. We specifically hypothesise that attacks may stem from the lack of compliance with high-level security measures—particularly those articulated in security directives.

This chapter focuses on the link between non-compliance and concrete attacks, namely the attack patterns defined in the CAPEC framework (Common Attack Pattern Enumeration and Classification) framework [93] as a reference. CAPEC provides a structured and system-agnostic classification of attack methods, enabling us to examine general exploitation strategies without being constrained to specific technologies or implementations.

We derive these attack patterns from the security measures outlined in the NIS 2 Directive. By mapping directive measures to CAPEC attack patterns, we aim to identify where non-compliance may result in

exposure to well-known classes of attacks. The hierarchical nature of the CAPEC taxonomy also allows for deeper technical analysis by linking attack patterns to more granular software weaknesses (CWEs) and publicly disclosed vulnerabilities (CVEs), thereby bridging the gap between abstract legal obligations and practical security risks

We designed WISARD (WIse Semantic(s) for Attack patteRns and Directives), a methodology composed of three steps. The first step consists of employing two *semantic methods* for correlating attack patterns with security measures. The semantic methods are based on the concepts of semantic similarity and ontological semantics, respectively. Semantic similarity uses statistical models to measure how closely two texts relate in meaning. Ontological semantics relies on structured knowledge bases to define meaning through formal relationships. The first semantic method introduces two families of algorithms designed to employ multiple similarity models and analyse the semantic similarity values applied to the security measures taken *verbatim* from the Directive. The objective is to identify convergent correlations across the multiple models. In contrast, the second semantic method employs prompt engineering, which is structured upon the grammatical structure of the security measures represented through ontological triples. Through specific prompts, we focus on different types of correlations that can exist between security measures and attack patterns. The objective of employing the ontological structure is to yield more granular correlations. The second step consists of creating a ground truth through the involvement of security experts to validate the correlations obtained with the semantic methods. However, significant divergence in expert responses led us to repurpose their output: rather than using it solely for validation, we use it to intersect with the correlations identified by semantic methods. The third step, therefore, consists of intersecting the correlations found by the semantic methods with expert feedback. By intersecting, we produce a knowledge base of correlations between security measures of the NIS 2 Directive and attack patterns from CAPEC that is validated by semantic methods and security experts.

WISARD combines the capabilities of human and artificial intelligence methods. The semantic methods, thanks to their intrinsic properties, understand a broader context than the sole knowledge of experts, which can be limited. However, WISARD still employs the human factor to confine and eventually correct the answers of the semantic methods, by leveraging the experience gained in the field by security experts. We can assume that if a correlation were found between an attack pattern and a security measure by semantic methods, this may not necessarily occur in

reality.

Furthermore, WISARD offers a more targeted and reliable approach compared to directly relying on general-purpose modern language models. Even though modern models can connect to the internet/websites and parse their content, they suffer from quantitative and qualitative problems. In quantitative terms, they may fail to retrieve comprehensive information; in qualitative terms, their outputs may not correctly represent the original content, producing artefacts. WISARD mitigates these limitations by specifically applying the semantic methods to the security domain and combining expert validation.

5.2 Related Work

Earlier research developed techniques to derive structured security knowledge from textual security documentation. Some approaches consider: NLP (Natural Language Processing) for extracting the relevant parts from security reports to create malware databases and predict the actions the malware performs [81]; NLP for extracting information from cyber-threat intelligence reports to be applied to malware analysis and search of inconsistencies in CVEs [105]; NLP and LLMS for inferring recovery steps from cyber threat intelligence reports, in the context of security operation centers [74]; semantic similarity and LLMs for mapping CVEs to CWEs [122]; ontological semantics for structuring security directives in a logical format [27]. The research question we are proposing in this chapter is part of a field that we observed is still unexplored. Unfortunately, we cannot consider it a rather active research field for which contributions already exist, especially in the context we are considering, of the NIS 2 Directive and the attack patterns from CAPEC. As far as we know, there are not many works that correlate security measures with attacks, especially considering text analysis functions such as semantic similarity in our case, or broadly speaking, in the context of AI-driven security [119]. For this reason, we present a series of works that deal with topics similar to ours, albeit not in the same direction, also covering the newest ML technologies.

Rongrat et al. [116] present a work where apart from the targets of the study being different from ours, the main similarity with ours stands in correlating both the notions of non-compliance (with banking security requirements) and attack patterns, although they have been used for a different study and in different CAPEC's field. If we used the CAPEC description field to know which attack correlates to the

security measure, the authors employed the Mitigations field to know the mitigations of non-employing security measures. Another remarkable difference is also in the approach since the association between the banking security requirements and the attack patterns is verified by manual validation. Kanakogi et al. [69] suggest using TF-IDF and Doc2Vec to automatically determine the associated CAPEC-IDs from CVE-ID, confirming experimentally that TF-IDF outperforms Doc2vec in terms of accuracy. Although the objective of the work is similar to ours in finding a correlation between two security fields using functions of text evaluation, they differ in such fields and the instruments, since we employ semantic similarity over LM and LLM models, and not TF-IDF. Vanamala et al. [136] introduce a technique to recommend CAPEC attack patterns by analysing software requirement specifications (SRS) through topic modelling. It matches topics from attack patterns and SRS documents, including descriptions and user requirements, using cosine similarity to determine relevance. The patterns are ranked, with the most pertinent suggested to developers for enhancing software defence. Li et al. [80] suggest a technique that utilises the CAPEC database to systematically model and implement attack patterns in Socio-Technical Systems. This method has been applied to 102 CAPEC patterns to semi-automatically detect and execute realistic attack strategies. Soltani et al. [125] examine actual links from cybersecurity graphs such as MITRE CAPEC, D3FEND, and CVE, and contrast them with forecasts made by large language models (LLMs) using semantic textual similarity (STS). The research incorporates various models including GPT-3.5, PaLM, and the specialised cybersecurity model ATTACK BERT. Aghei et al. [8] introduce the TTPpredictor tool, which deduces potential TTP attacks from CVE descriptions. Utilising Semantic Role Labelling, the tool extracts threat actions from cyber threat reports and aligns them with MITRE's attack categories to produce labelled data, leading to accurate CVE classification and improved proactive threat management. Lee et al. [79] confront the escalating issue of security vulnerabilities in software and hardware systems, challenging the Common Vulnerability Scoring System (CVSS) for its narrow evaluation of vulnerability characteristics. They suggest a semantic ranking approach that evaluates vulnerabilities by examining their relational information—essentially, how similar or connected they are to each other. Abdeen et al. [7] introduce SMET, a tool that automatically connects CVE entries to ATT&CK techniques using textual similarity. SMET employs ATT&CK BERT, a model the authors developed with a SIAMESE network to understand semantic similarities among attack actions. During inference, SMET uses semantic extraction,

ATT&CK BERT, and a logistic regression model to associate CVE entries with specific ATT&CK techniques. Akbar et al. [9] utilise cutting-edge machine learning technologies to offer a robust solution for security analysts. The proposed tool employs sophisticated natural language processing methods, such as a large language model (RoBERTa), to extract significant semantic links between descriptions of attack methods and defensive responses.

5.3 Special Focus On Attack Patterns And Semantic Similarity

This Section presents the fundamental concepts behind the chapter. We present the inputs we analysed, the NIS 2 Directive and the attack patterns (together with the CAPEC framework), and the mathematical notion of semantic similarity.

NIS 2 Directive We have already discussed NIS 2 at length, so in this case, we will limit ourselves, in an extract in Table 5.1, to highlighting some measures that demonstrate the lack of specificity and detail that security operators will have to deal with.

Table 5.1: Extract of the NIS 2 Directive

Chapter	Article	Paragraph	Security Measure
2	7	1	Each Member State shall designate or establish one or more competent authorities responsible for the management of large-scale cybersecurity incidents and crises...
2	21	1	Member States shall ensure that essential and important entities take appropriate and proportionate technical, operational and organisational measures...
2	21	2(d)	The measures shall include [...] supply chain security [...]
2	21	2(h)	The measures shall include [...] policies and procedures regarding the use of cryptography and, where appropriate, encryption
...	...	...	...

These succinct examples show that, despite the measures being taken from what are generally considered to be more technical articles, the actual level of detail included in these measures is still fairly low and frequently lacks the specificity needed for clear operationalisation.

This implies that security-related instructions may still be conveyed in ambiguous or general words even in supposedly technical circumstances, which could make it more difficult to understand and apply them effectively.

Attack Patterns Attack Patterns are the common techniques used by attackers to exploit known weaknesses in cyber-enabled technologies. Each attack pattern contains details about the particulars of how an attack is organised and executed. Attack Patterns represent the highest, and therefore most abstract, step of a hierarchy composed of Attack Patterns, CWEs and CVEs, therefore there is a natural correlation between the concepts. An attack pattern can specialise in multiple CWEs, and subsequently, each CWE materialises in a CVE when the CWE is found in a specific version of a commercial product/service. The attack patterns are classified in a framework called CAPEC [93] maintained by MITRE. CAPEC contains 559 attack patterns, each one described by several attributes. In our research, we used the CAPEC ID, the CAPEC Description, and the Related CWE (each one stands for, respectively, the attack pattern ID, the attack pattern description, and the attack pattern's related CWE). An extract of how the CAPEC framework is composed is illustrated in Table 5.2.

Table 5.2: Extract of the CAPEC Framework

ID	Name	Description	Related CWEs	...
98	Phishing	Phishing is a social engineering technique...	451	...
100	Overflow Buffers	Buffer Overflow attacks target improper or missing bounds checking on buffer operations...	119, 120, ...	...
...	...	...	...	...

Semantic Similarity To correlate the security measures derived from the NIS 2 Directive and the attack patterns derived from CAPEC, we need a calculation that mainly considers the description and semantic meanings of both. Comparing semantic meanings is the only way to proceed since the security measures are not intrinsically correlated with any other security concepts. Therefore, we approached the problem by considering the *semantic similarity*, i.e., how close the meanings of words, sentences, and texts are. A way to calculate the semantic similarity is by considering the *cosine similarity*, i.e., a mathematical measure that calculates the cosine of the angle between the vectors. Given two numerical vectors A and B, cosine similarity is a mathematical function calculated over the two vectors, according to the following formula (and its extended version).

$$\text{Cosine Similarity } (A, B) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \|\vec{B}\|}$$

$$\text{Cosine Similarity } (A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Where:

- $\vec{A} \cdot \vec{B}$: Dot product of vectors $\vec{A}$ and $\vec{B}$.
- $\|\vec{A}\|$: Magnitude of vector $\vec{A}$, calculated as $\|\vec{A}\| = \sqrt{\sum_i A_i^2}$.
- $\|\vec{B}\|$: Magnitude of vector $\vec{B}$, calculated as $\|\vec{B}\| = \sqrt{\sum_i B_i^2}$.

Range: If vectors are non-negative, the similarity is in the range $[0, 1]$. Otherwise, the similarity is in the range $[-1, 1]$.

For applying semantic similarity over two input it is therefore necessary to transform the strings into numerical vectors so that the proper cosine similarity can be calculated.

The strings are represented as numerical vectors in a continuous vector space via the *word embeddings*.

$$\vec{w}_i \in \mathbb{R}^d \quad \text{for} \quad i = 1, 2, \ldots, n$$

Where $\vec{w}$ is the embedding of the word w, and d is the dimensionality of the embedding space. A very common approach in the scientific community is to leverage pre-trained models to obtain the embeddings where the weights of the words are given, considering the data with which the models have been trained. Each model, therefore, if accurately chosen, can represent a *slightly different perspective* about the interpretation of semantic meanings. How two words are translated into embeddings is illustrated with the following examples.

$$Malware_X: \begin{bmatrix} 0.8 \\ 0.5 \\ 0.2 \end{bmatrix} \qquad Ransomware_X: \begin{bmatrix} 0.9 \\ 0.4 \\ 0.1 \end{bmatrix}$$

$$Malware_Y: \begin{bmatrix} 0.6 \\ 0.5 \\ 0.4 \end{bmatrix} \qquad Ransomware_Y: \begin{bmatrix} 0.8 \\ 0.5 \\ 0.2 \end{bmatrix}$$

Considering different models, X and Y, the words *Malware* and *Ransomware* assume a different embedding, i.e., a different numerical vector representing the word itself in the multidimensional space of the model.

5.4 The WISARD Methodology

We propose WISARD, a methodology consisting of three steps, combining human and artificial intelligence capabilities. In the first step, the *Semantic step*, we apply two automated methods relying on the concept of *semantics* for correlating the security measures with the attack patterns. In the second step, the *Validation Base Construction step*, we construct a validation base by employing the knowledge of security experts. In the third step, the *Intersection step*, we intersect the correlations obtained from the Semantic step with the validation base from the Validation Base Construction step.

WISARD is illustrated in Figure 5.1 and, in more detail, the steps are designed as follows.

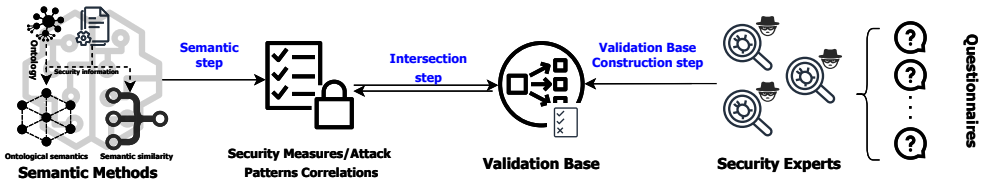


Figure 5.1: Illustration of WISARD

Semantic step The purpose of the Semantic step is to apply semantic methods for mapping the security measures to attack patterns.

In the Semantic step, we implemented two methods, both based on the concept of semantics, although differing in their specific techniques. The first method is based on *semantic similarity*, hence termed *Semantic Similarity step*. By designing two families of algorithms that compute different semantic similarity values obtained by applying different transformer models, this method evaluates the presence and the degree of correlation occurring between a measure and an attack pattern.

The second method is based on *ontological semantics*, hence termed *Ontological Semantics step*. Building on `SecOnto`, hence on `NIS2Onto`, the method leverages the deconstruction of the security measures in a logical format, called triples, containing the grammatically most relevant element of each measure. The triples are then used to perform targeted correlations with attack patterns utilising prompt engineering techniques typical of LLMs. The key distinction lies in the approach: the first method leverages each measure as a whole statement, while the second leverages the dissection of each measure into logical components for more fine-grained analysis.

Validation Base Construction step The purpose of the Validation Base Construction step is to provide a validation base through a collective agreement between security experts.

In the Validation Base Construction step, the security experts are asked to agree on which attack patterns are most relevant to each security measure. This process constitutes *the creation of ground truth*, a collective and objective validation base designed upon expert consensus, which can be used to test the semantic methods.

Nonetheless, in the security field, there is currently no ground truth that correlates measures derived from any security directive and attack patterns. To address this gap, the Validation Base Construction establishes a ground truth using structured questionnaires. The questionnaires were administered to security experts, who were tasked of finding correlations between security measures and attack patterns.

Intersection step The purpose of the Intersection step is to validate the semantic methods using the validation base.

In this step, we intersect the correlations resulting from the application of the semantic methods with the validation base designed from the questionnaires, obtained by the security experts. In doing so, we build a mapping both algorithmically and by expert consensus between security measures written in legal language and attack patterns.

In the next three sections, we will show the Semantic step, which applies the semantic methods (divided into the Semantic Similarity step and Ontological Semantics step), the Validation Base Construction step of building the validation base through the employment of security experts, and the Intersection step for intersecting the findings of the prior steps. Each section consists of a subsection analysing the method applied for

the specific step and the purpose behind it, and a subsection discussing the application of the same.

For specific criteria that will be discussed in the Validation Base Construction step, the semantic methods are applied to Articles 7 and 21 of the NIS 2 Directive.

5.5 Semantic Similarity Step

The Semantic Similarity step consists of correlating security measures and attack patterns, leveraging the concept of *semantic similarity*, i.e., obtaining meaningful information.

The following subsections present two algorithms that leverage the concept of *collective convergence*, i.e., employing multiple transformer-based models for aggregating insights.

5.5.1 Method for the semantic similarity step

The Semantic Similarity step is composed of the following sub-steps:

1. **Computational sub-step**. The sub-step consists of simply cataloguing the inputs and calculating the cosine similarity.
2. **Correlational sub-step** The sub-step consists of providing heuristic algorithms that, starting from the cosine similarity results, are able to find objective correlations between security measures and attack patterns.

5.5.1.1 Computational sub-step

The **Computational sub-step** consists of selecting the datasets D1 and D2 from where the texts will be taken as inputs, and applying the cosine similarity for each couple d1, d2, where d1 $\in$ D1 and d2 $\in$ D2, belonging to the cartesian product D1 $\bigtimes$ D2;

Since the cosine similarity is applied over N number of models to the cartesian product D1 $\bigtimes$ D2, the resulting computation will produce a dataset of cardinality |D1 $\bigtimes$ D2| x N. In our work, D1 is composed of the descriptions of all the attack patterns, each attack pattern identified with the notation $AP1, AP2, AP3, \ldots, AP|D1|$, while D2 is composed of the NIS 2 measures specified in the Validation Base Construction step, each measure identified with the notation $M1, M2, M3, \ldots, M|D2|$.

Table 5.3 partially illustrates how the resulting semantic similarity (SS) values for D1 and D2 and each model are placed in a proper dataset.

Table 5.3: Extract of Example Dataset

AP	Measure	Model 1	Model 2	...	Model N
AP1	M1	$SS_{(1,1,1)}$	$SS_{(1,1,2)}$	...	$SS_{(1,1,N)}$
AP2	M1	$SS_{(2,1,1)}$	$SS_{(2,1,2)}$	...	$SS_{(2,1,N)}$
AP3	M1	$SS_{(3,1,1)}$	$SS_{(3,1,2)}$	...	$SS_{(3,1,N)}$
...	...	...	...	...	...
AP1	M2	$SS_{(1,2,1)}$	$SS_{(1,2,2)}$	...	$SS_{(1,2,N)}$
AP2	M2	$SS_{(2,2,1)}$	$SS_{(2,2,2)}$	...	$SS_{(2,2,N)}$
AP3	M3	$SS_{(3,3,1)}$	$SS_{(3,3,2)}$	...	$SS_{(3,3,N)}$
...	...	...	...	...	...

The entire table serves as input for the algorithms used in the second step of the method, namely the **Correlational sub-step**. In our assumption, each model chosen for the Computational sub-step represents a different security analyst, interpreting the inputs according to their vision. With this approach, we simulate that each model produces a different embedding and that using different models to calculate semantic similarity allows us to obtain a heterogeneous view of the associations between attack patterns and measures. The heterogeneity of values is fundamental to understanding whether these different perspectives converge in a correlation.

5.5.1.2 Correlational sub-step

The **Correlational sub-step** is designed to find correlations between attack patterns and security measures. The Correlational sub-step consists of applying a heuristic that, given the semantic similarity values calculated in the Computational sub-step as input, can automatically discern which association attack patterns and measures are correlated. This heuristic is implemented by two designed families of algorithms that rely on the concept of max value. The approach consists of a completely automatic way of finding a convergence heuristic by solely analysing the semantic similarity values produced by all the described models. The multitude of models will be leveraged to evaluate the local perspectives, to assess their global convergence. The individual calculated values can be insightful, but only reflect the perspective of a single model. Therefore, if the model is biased or hallucinated, the single values could negatively inflate the valuation. The multi-model approach is designed for only considering the correlations confirmed by all the models, or a minimal set, to smooth

possible outliers, i.e., models that are introducing wrong correlations (Table 5.4).

Table 5.4: Extract of Example Dataset

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP1	M1	$SS_{(1,1,1)}$	$SS_{(1,1,2)}$	...	$SS_{(1,1,N)}$	✔
AP2	M1	$SS_{(2,1,1)}$	$SS_{(2,1,2)}$	...	$SS_{(2,1,N)}$	✔
AP3	M1	$SS_{(3,1,1)}$	$SS_{(3,1,2)}$	...	$SS_{(3,1,N)}$	✘
...	...	...	...	...	...	...
AP1	M2	$SS_{(1,2,1)}$	$SS_{(1,2,2)}$	...	$SS_{(1,2,N)}$	✔
AP2	M2	$SS_{(2,2,1)}$	$SS_{(2,2,2)}$	...	$SS_{(2,2,N)}$	✘
...	...	...	...	...	...	...

The rationale behind fully automating the correlation process derives from the notion of semantic similarity itself, indicating that the values closest to 1 suggest a high probability of similarity between the two given inputs. However, determining a meaningful threshold below 1 to obtain a correlation is challenging, since there is no objective criterion. Therefore, we introduce the concept of ***max value*** (hereafter simply *max*). As implied by the term, a max represents the highest value of semantic similarity obtained with a particular model, i.e., a max is specific to a model; it is not the highest value across all models. As a consequence, only the attack pattern/measure pairs that exhibit high semantic similarity values across the models should be chosen to, in the end, identify the correlated ones.

Considering these assumptions, we present two families of algorithms: **AMax** (*Absolute Max*), and **RMax** (*Relative Max*). AMax algorithms are based on the concept of *Absolute Max*, i.e., an absolute max value shared among the models; RMax algorithms are based on the concept of *Relative Max*, i.e., relative max values distributed among the models. The two families are divided into main algorithms, AMax0 and RMax0, and AMax1 and RMax1, as more generalised variants (correlations are accepted if supported even by a minimal subset of semantic similarity models) of AMax0 and RMax0, respectively.

$$|min(SSModels)| \geqslant \left\lfloor \frac{|SSModels|}{2} \right\rfloor, \quad where \quad |SSModels| \ > 2$$

Naturally, this is valid until a single model yields the max value, since the max in a single model will be present within each model. However, having a single maximising model does not hold any meaningful significance, since it would not allow for catching the general trend. Moreover, at least one max value among all computed similarities is always guaran-

teed. Therefore, the number of models must be greater than 2 to ensure a meaningful convergence.

We first introduce the two families in their 1-to-1 version, in which an attack pattern is associated with a security measure. Subsequently, we present a 1-to-Many version of AMax0 and RMax0, or recursive version, where an attack pattern may be correlated with multiple security measures. As the recursive design for AMax1 and RMax1 is the same as AMax0 and RMax0, the 1-to-Many version of AMax1 and RMax1 will be omitted. Each algorithm will be demonstrated on an illustrative and finite set of computed semantic similarity values, which may not necessarily reflect real-world correlations between the attack pattern and the measure of the same row.

The following paragraphs detail the descriptions of the two families of algorithms in both versions. For each case, we illustrate the rationale behind the algorithm, the pseudocode, and a tabular example, followed by a brief analysis of the obtained results. Across all pseudocode cases, the set variables are: Semantic Similarity Models (*SSModels*), Security Measures (*CM*), and Attack Patterns (*AP*).

AMax 1-to-1 The 1-to-1 version of the AMax family has the following rationale: *given an attack pattern, find the measure where the models share the same max value.* In the case of AMax0, ***all*** the models should be maximising.

The consideration behind AMax is that if all the models have the same semantic similarity value, they converge towards the same direction, and if the value is max, it strongly indicates that they converge towards a correlation between an attack pattern and a measure.

The pseudocode algorithm for the 1-to-1 Version of AMax0 is illustrated in Algorithm 1.

3-4 Initialise a loop for iterating over all the attack patterns. Initialise a variable for counting the number of maximising models;

5-6 Iterate through the models for finding the max value of semantic similarity. The max value will be in a specific model and a specific measure;

7 Check if the found max is greater than the max already stored, and update it;

8 Find and store the measure where the updated max is;

Algorithm 1: AMax0 — 1-to-1

```
   Inputs: Semantic Similarity Models SSModels; Non-Compliant Measures CM; Attack
           Patterns AP
1
2  Function retrieve_max(AP, SSValues, SSModels):
3      for ap ∈ AP do
4          max_models ← ∅
5          for mo ∈ SSModels do
6              max ← find_max(ap, mo)
7              if update_max(max) then
8                  measure_(ap,max) ← find_measure(max_(ap,mo))
9
10         for mo ∈ SSModels do
11             if find_max(ap, mo, measure_(ap,max)) == max then
12             max_models ← max_models ∪ mo
13         if |max_models|== |SSModels| then
14         collect(ap, measure_(ap,max), max, max_models)
15     return dataset
```

10-12 Find and store the models that have the previously found max, in the measure where the max is;

13-14 Check if the number of models sharing the max value is equal to the cardinality of models. If the check is passed, then store the measure the attack pattern, the measure, the max and models where the max has been found.

Applying AMax0 over N models would be as illustrated in Table 5.5.

Table 5.5: Correlation Example with AMax0

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP1	M1	0.15	0.21	...	0.31	✘
AP1	M2	0.08	0.22	...	0.16	✘
AP1	M3	0.42	0.42	...	0.35	✘
AP1	M4	0.8	0.8	...	0.8	✔
AP1	M5	0.16	0.31	...	0.39	✘
AP1	M6	0.25	0.21	...	0.2	✘
AP1	M7	0.17	0.22	...	0.3	✘

Applying AMax0 in the previous dataset will correlate AP1 and M4 since the condition that all models have the same max value in a single measure is respected.

The case of the AMax1 version arises if we decide to relax the hypothesis of having all the models resulting in being maximising. We can then decrease the number of models on which we would like to retrieve the

max value until we meet the *minimal set* of maximising models based on the conditions discussed before. The cardinality of the minimal set could be chosen by a security analyst who is in charge of deciding the number of correlations they would like to handle, being that naturally having a relaxation in the condition will simultaneously have a lower precision.

The pseudocode algorithm for the 1-to-1 Version of AMax1 is illustrated in Algorithm 2. Algorithm 2, in line 13, shows how the absoluteness condition for all models of the AMax0 algorithm is approached considering a minimal set.

Algorithm 2: AMax1 — 1-to-1

```
   Inputs: Semantic Similarity Models SSModels; Non-Compliant Measures CM; Attack
           Patterns AP
1
2  Function retrieve_max(AP, SSValues, SSModels):
3      for ap ∈ AP do
4          max_models ← ∅
5          for mo ∈ SSModels do
6              max ← find_max(ap, mo)
7              if update_max(max) then
8                  measure_(ap,max) ← find_measure(max_(ap,mo))
9
10         for mo ∈ SSModels do
11             if find_max(ap, mo, measure_(ap,max)) == max then
12                 max_models ← max_models ∪ mo
13         if |max_models|>= min(SSModels) then
14         collect(ap, measure_(ap,max), max, max_models)
15     return dataset
```

Applying AMax1 over N models, considering a decreasing set of minimal models, would be like Table 5.6 and Table 5.7.

Table 5.6: Correlation Example with AMax 1 — N-1 Models

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP1	M1	0.15	0.21	...	0.31	✘
AP1	M2	0.08	0.22	...	0.16	✘
AP1	M3	0.42	0.42	...	0.35	✘
AP1	M4	0.57	0.57	...	0.48	✔
AP1	M5	0.16	0.31	...	0.39	✘
AP1	M6	0.25	0.21	...	0.2	✘
AP1	M7	0.17	0.22	...	0.3	✘

Table 5.7: Correlation Example with AMax 1 — reached the minimum set

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP1	M1	0.15	0.21	...	0.31	✘
AP1	M2	0.08	0.22	...	0.16	✘
AP1	M3	0.42	0.42	...	0.35	✘
AP1	M4	0.57	0.57	...	0.48	✔
AP1	M5	0.16	0.31	...	0.39	✘
AP1	M6	0.25	0.21	...	0.2	✘
AP1	M7	0.17	0.22	...	0.3	✘

RMax 1-to-1 The 1-to-1 version of RMax has the following rationale: *given an attack pattern, find the measure where the models have their max value*. In the case of RMax0, ***all*** the models should be maximising, hence each model obtained a max value in the measure, and the max value is different among the models.

As opposed to the AMax family, in the RMax family, we have loosened the constraint of having identical max values, but we have established that each model must possess its max value. This means that, although the models may not share the same max value, it is still essential to confirm that each model has a max value, since, even in this case, it is a strong indicator that the models converge towards a correlation between a security measure and an attack pattern.

The pseudocode algorithm for the 1-to-1 Version of RMax0 is illustrated in Algorithm 3.

Algorithm 3: RMax0 — 1-to-1

```
   Inputs: Semantic Similarity Models SSModels; Non-Compliant Measures CM; Attack
           Patterns AP
1
2  Function retrieve_max(AP, SSValues, SSModels):
3      for ap ∈ AP do
4          extracted_measures ← ∅
5          for mo ∈ SSModels do
6              max ← find_max (ap, mo)
7              measure_(ap,max) ← find_measure (max_(ap,mo))
8              extracted_measures ← extracted_measures ∪ measure_(ap,max)
9
10         if |Models_extracted_measures| == |SSModels| then
11         collect(ap, extracted_measures[0], max)
12     return dataset
```

3-4 Loop through the attack patterns, and initialise a variable for storing the measures where the max values will be found;

5-8 Loop through the models. For each model, find the measure where the max value between the measure and the attack pattern is, and store the measure;

10-11 Check if the cardinality of the extracted measures is equal to the cardinality of the models. In this case, all the models have their max in the same measure. Then collect the attack pattern, the single measure and the max;

Applying AMax0 over N models would be as illustrated in Table 5.8.

Table 5.8: Correlation Example with RMax0

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP2	M1	0.11	0.22	...	0.78	✘
AP2	M2	0.12	0.29	...	0.74	✘
AP2	M3	0.64	0.66	...	0.84	✔
AP2	M4	0.4	0.54	...	0.81	✘
AP2	M5	0.23	0.31	...	0.77	✘
AP2	M6	0.19	0.2	...	0.76	✘
AP2	M7	0.19	0.24	...	0.76	✘

As can be seen in Table 5.8, every model in the context of attack pattern AP2 has the max value in the same measure, which is measure M3, since the condition that all models obtain their own max value in a single measure is respected.

Instead, for the RMax1 version, the same considerations analysed for AMax1 apply. The pseudocode algorithm for the 1-to-1 Version of RMax1 is illustrated in Algorithm 4. Algorithm 4, in line 11, shows how the absoluteness condition for all models of the RMax0 algorithm is approached considering a minimal set.

Applying RMax1 over N models, considering a decreasing set of minimal models, would be like Table 5.9 and Table 5.10.

Table 5.9: Correlation Example with RMax1 — N-1 Models

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP2	M1	0.11	0.26	...	0.78	✘
AP2	M2	0.12	0.65	...	0.74	✘
AP2	M3	0.64	0.62	...	0.84	✔
AP2	M4	0.4	0.48	...	0.81	✘
AP2	M5	0.23	0.4	...	0.77	✘
AP2	M6	0.19	0.22	...	0.76	✘
AP2	M7	0.19	0.36	...	0.76	✘

Algorithm 4: RMax1 — 1-to-1

Inputs: Semantic Similarity Models SSModels; Non-Compliant Measures CM; Attack Patterns AP

```
1
2  Function retrieve_max(AP, SSValues, SSModels):
3      for ap ∈ AP do
4          extracted_measures ← ∅
5          for mo ∈ SSModels do
6              max ← find_max (ap, mo)
7              measure_(ap,max) ← find_measure (max_(ap,mo))
8              extracted_measures ← extracted_measures ∪ measure_(ap,max)
9
10         intersection (extracted_measures)
11         if |Models_extracted_measures| >= min(SSModels) then
12         collect(ap, extracted_measures[0], max)
13     return dataset
```

Table 5.10: Correlation Example with RMax1 — reached the minimum set

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP2	M1	0.22	0.26	...	0.78	✘
AP2	M2	0.29	0.65	...	0.74	✘
AP2	M3	0.66	0.62	...	0.84	✔
AP2	M4	0.54	0.48	...	0.81	✘
AP2	M5	0.31	0.4	...	0.77	✘
AP2	M6	0.2	0.22	...	0.76	✘
AP2	M7	0.24	0.36	...	0.76	✘

AMax 1-to-Many & RMax 1-to-Many To associate an attack pattern with multiple measures, the families of algorithms AMax and RMax are presented in their 1-to-Many version.

In the 1-to-1 version, each attack pattern is exclusively associated with only one security measure; the algorithms iterate only once. Instead, in the 1-to-Many version, an attack pattern is evaluated against one measure at a time, after which the algorithms recursively iterate the heuristic to find if other measures can be correlated. To identify other measures, the algorithms search for a secondary ***max level***, distinct from the level obtained previously. The key conceptual difference would consist in discarding the values where the previous correlation was obtained. Through the recursion, the algorithms seek in depth if another measure satisfies the same requirements. In the best case, the recursion will iterate $|D2|$ times.

In the 1-to-Many version, the terminating condition of the recursion consists of the impossibility of finding certain requirements, which are

peculiar to each family of algorithms: in the AMax family the number of maximising models is greater than 1 (leading to the terminating condition if the number of maximising models is lower than 2); in Algorithm 2 the convergence in a single measure (leading to the terminating condition if the intersection is in more than one measure).

In terms of pseudocode, the modification is illustrated in the grey box in the algorithms 5 6, and the same analysis applies to both algorithms. The execution of the 1-to-Many version of the algorithms is presented via illustrative examples in Table 5.11 and Table 5.12, respectively, for AMax0 and RMax0. In each table, the darker green colour of the cells gradually becomes lighter in case new correlations in depth are found.

The main difference between the pseudocodes 1 and 3 is the removal of the outer loop that iterates through all the attack patterns, to apply the recursion on the same attack pattern.

The algorithm for the 1-to-Many Version of AMax0 is illustrated in Algorithm 5.

Algorithm 5: AMax0 — 1-to-Many

```
   Inputs: Semantic Similarity Models SSModels; Non-Compliant Measures CM; Attack
           Patterns AP
1
2  Function retrieve_max(ap, SSValues_ap, SSModels):
3      max_models ← ∅
4      for mo ∈ SSModels do
5          max ← find_max(ap, mo)
6          if update_max(max) then
7          measure_(ap,max) ← find_measure(max)
8
9      for mo ∈ SSModels do
10         if find_max(ap, measure_(ap,max), mo) == max then
11         max_models ← max_models ∪ mo
12         collect(ap, measure_(ap,max), max)
13
14     if |max_models|== |SSModels| then
15         remove(SSValues_ap, measure_(ap,max))
16         retrieve_max(ap, SSValues_ap, SSModels)
17     else return dataset
18
```

14 Check if the number of max models is different from one;

15 Remove all the semantic similarity values related to the attack pattern and the measure where the max was found;

15 Recursively run the algorithm, else return the values.

Table 5.11: Correlation Example with AMax1 — 1-to-Many

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP3	M1	0.08	-0.03	...	0.17	✘
AP3	M2	0.04	0.14	...	0.06	✘
AP3	M3	0.11	-0.04	...	0.15	✘
AP3	M4	0.25	0.25	...	0.25	✔
AP3	M5	0.01	-0.02	...	0.19	✘
AP3	M6	0.19	0.19	...	0.19	✔
AP3	M7	0.06	-0.12	...	0.16	✘

Instead, for the 1-to-Many version of RMax, we have the following considerations...

The pseudocode algorithm for the 1-to-Many Version of RMax0 is illustrated in Algorithm 3.

Algorithm 6: RMax0 — 1-to-Many

Inputs: Semantic Similarity Models SSModels; Non-Compliant Measures CM; Attack Patterns AP

```
1
2  Function retrieve_max(ap, SSValues_ap, SSModels):
3      extracted_measures ← ∅
4      for mo ∈ SSModels do
5          max ← find_max (ap, mo)
6          measure_(ap,max) ← find_measure (max)
7          extracted_measures ← extracted_measures ∪ measure_(ap,max)
8
9      intersection (extracted_measures_values)
10
11     collect(ap, extracted_measures[0], max)
12
13     if |Models_extracted_measures|== |SSModels| then
14         remove(SSValues_ap, extracted_measures[0])
15         retrieve_max(ap, SSValues_ap, SSModels)
16     else return dataset
17
```

13. Check if the number of extracted measures is one;

14. Remove all the semantic similarity values related to the attack pattern and the measure where the max was found;

15. Recursively run the algorithm, else return the values.

Table 5.12: Correlation Example with RMax1 — 1-to-Many

AP	Measure	Model 1	Model 2	...	Model N	Correlation
AP4	M1	0.13	0.15	...	0.74	✘
AP4	M2	-0.03	0.04	...	0.71	✘
AP4	M3	0.27	0.2	...	0.75	✔
AP4	M4	0.34	0.31	...	0.78	✔
AP4	M5	0.15	0.15	...	0.74	✘
AP4	M6	0.2	0.09	...	0.74	✘
AP4	M7	0.11	0.11	...	0.73	✘

5.5.2 Applying the semantic similarity step

To apply the Semantic Similarity step, we made two types of key choices. The first type consists of selecting different models for embedding calculation: all-Minilm-L6-v2, attack-bert, all-mpnet-base-v2, and paraphrase-multilingual-mpnet-base-v2. All-MiniLM-L6-v2 and all-mpnet-base-v2 are among the best-performing general models according to the STS benchmarks; paraphrase-multilingual-mpnet-base-v2 extends similarity modelling to multilingual settings; attack-bert is a domain-adapted model trained on cyber threat intelligence data.

A graphical representation of the embeddings calculated on a subset of measures, as the 4 models vary (in the same order as described), is shown in the following figures (Figure 5.2, Figure 5.3, Figure 5.4, and Figure 5.5).

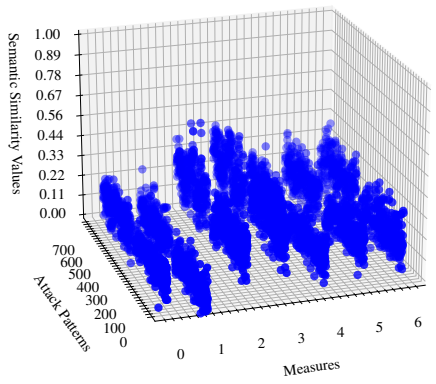


Figure 5.2: Cosine similarity between attack patterns and a subset of security measures – all-Minilm-L6-v2 model

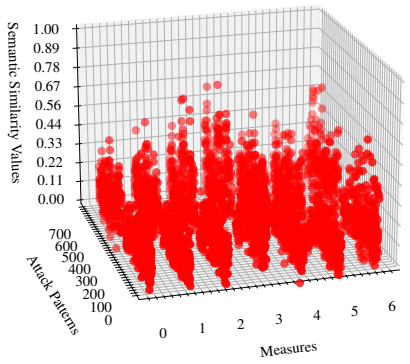


Figure 5.3: Cosine similarity between attack patterns and a subset of security measures – attack-bert

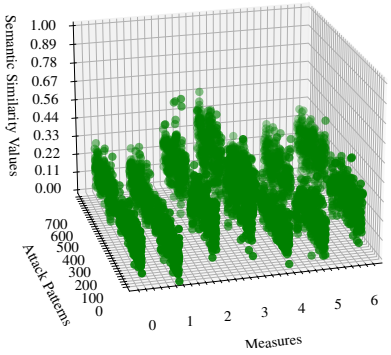


Figure 5.4: Cosine similarity between attack patterns and a subset of security measures – all-mpnet-base-v2

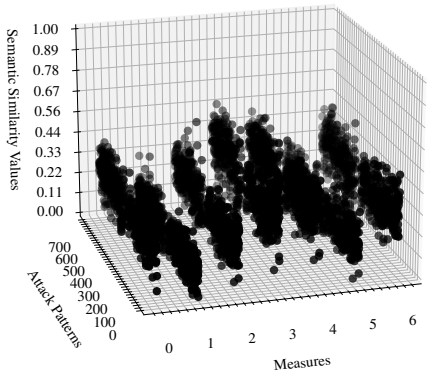


Figure 5.5: Cosine similarity between attack patterns and a subset of security measures – paraphrase-multilingual-mpnet-base-v2

The second type consists of varying the input D1. Specifically, semantic similarity was computed both on the original security measures from articles 7 and 21, and on extended versions of those measures. The choice stems from the observation that the measures, taken in their original form, are rather general and non-specific. Finding an extension means having a measure that is more complete and that expands, in an agnostic way, the content of its original version. To generate these extensions, we consulted 3 different language models: LLama3.2, Phi4 and Mistral. So, each single measure will have 3 different versions, each provided by the chosen model. These models were deliberately selected on the basis of their established reputation in the industry, their architectural and functional diversity and their proven effectiveness. For the specific task of extending a security measure, their combined capabilities were deemed sufficiently comprehensive to ensure robust and representative results.

The settings employed for the Semantic Similarity step and the number of correlations found between security measures and attack patterns are illustrated in Tables 5.19 and 5.20, respectively, for Articles 7 and 21 of the NIS 2 Directive. In particular, the column *N.Corr.* shows the Number of Correlations obtained using the specific setting of each row, under the column *Setting* — for the Semantic Similarity step, consider only the setting where the acronym **SS** appears.

5.6 Ontological Semantics Step

The Ontological Semantics step consists of correlating attack patterns to security measures by leveraging the *ontological semantics* of the measures, i.e., their formal ontological representation. The ontological representation proves particularly valuable for the prompt engineering task. By exploiting the ontological structure of each security measure, we can formulate targeted prompts to language models tailored to elicit correlations between attack patterns and measures.

Since the ontological representation of the security measures belonging to security directives has been previously addressed by the same authors [27], the following subsections show the method for handling *prompt engineering*, intending to accurately obtain attack patterns by harnessing the ontological perspective.

5.6.1 Method for the ontological semantics step

For the Ontological Semantics step, we adopt the concept of an AI agent. The agent operates with two key characteristics: a predefined sequence of tasks executed in order, and an internal state that is incrementally updated by each task. The agent state serves as a structured memory storing security measures and attack patterns, and most importantly, ensures consistency throughout the entire correlation process.

By leveraging this agent-based paradigm, the Ontological Semantics step is articulated into a set of sub-steps, illustrated in Figure 5.6.

1. **Selection**. This sub-step consists of selecting from the Directive the security measures that the analyst wants to correlate with the attack patterns.

2. **Security Measures Retrieval**. This sub-step consists of retrieving the chosen security measures from a given ontology. The operation is performed by using the SPARQL queries, as the standard mechanism for querying ontologies. Once retrieved, the selected security measures are stored in the internal state of the agent for subsequent processing.

3. **Attack Patterns Retrieval** This sub-step consists of extracting all the attack patterns from a specified source file. Unlike traditional parsing methods, the retrieval is performed using a language model. Once retrieved, the attack patterns are stored in the internal state of the agent.

4. **Final Answer Generation** This sub-step, as the final step, consists of leveraging the knowledge capabilities of the employed model to retrieve all the information from the second and third steps. Specifically, it must correlate the security measures from the ontology and the attack patterns, leveraging its capabilities. and providing possible motivations for the found correlations.

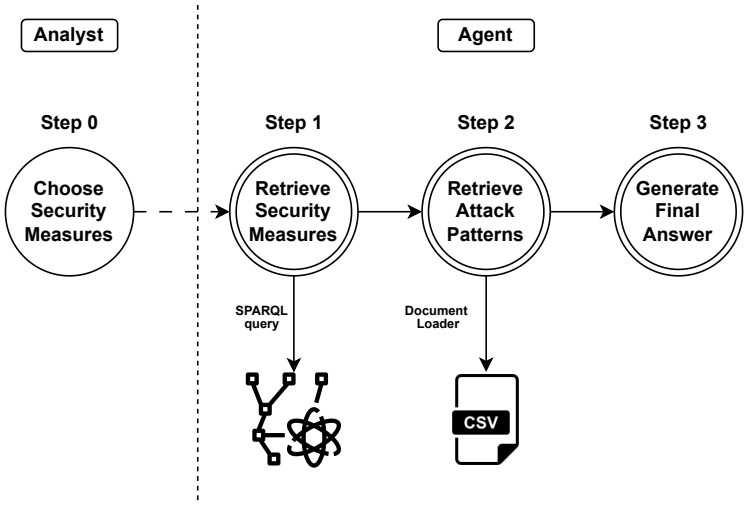


Figure 5.6: Graph of steps of the ontological semantics method

5.6.1.1 Selection

This sub-step involves the intervention of a security expert responsible for choosing the security measures from the Directive, hence from the ontology. Although the task itself is relatively simple, we highlight it since it represents the sole stage that necessitates direct human involvement. The next sub-steps are completely executed by the AI agent, which computes the assigned tasks. At the end of all the sub-steps, the analyst will obtain the results computed by the agent. As in the Semantic Similarity step, the selected measures correspond to those previously introduced in the Validation Base Construction step.

5.6.1.2 Security measures retrieval

This sub-step consists of retrieving the security measures from a given ontology. Although we have discussed the ontological question exten-

sively previously, for ease of reading, in this section, we provide a brief recap.

Ontologies, in their broader meaning, allow designers to provide a conceptual model of a specific domain, e.g., in this case, the domain of the NIS 2 Directive. Having an ontology of the NIS 2 means that we have a representation of the Directive content. The Directive is a set, more or less structured, of what we are calling security measures. Within the ontology, the security measures are structured through basic constructs called *triples*. A triple is formed by the elements, in this order: entity, property, and entity. The goal of a triple is to express the *relation* (property) that correlates the *first entity* with the *second entity*. The relation can be of different types. In the case of the Directive, we are stating that the property is the action that the Agent (first entity) of a security measure must fulfill towards the object (second entity) of the security measure. A straightforward example is the following: having the measure *Member State should adopt the National Cybersecurity Strategy*, it can be structured in an ontological triple (MemberState, adopt, NationalCybersecurityStrategy). We can consider a triple as a small building block, which, combined with other triples and using more complex operations, determines a conceptual modelling of the domain in an effectively structured way. Consequently, having an ontology-driven approach means extracting meaningfully and precisely the triples related to the specific entities we are referring to.

For this step, we designated a specific SPARQL query that can retrieve the desired security measures from the ontology, e.g., the security measures provided within a given Article of the Directive. The ontology we are using has been designed to represent all the security measures inside the NIS 2 Directive, and consequently, the ontological security measures tend to be highly structured, reflecting the formal complexity of the document. As a consequence, the SPARQL query must be robust enough to handle the nested cases and consider the ontological constructs used for building the ontological measures. In addition, all these elements can be found in considerable hierarchical depth, which requires traversing several hierarchical levels.

After retrieving the security measures related to a specific agent in a specific Article of the Directive, we store them in a local file for practical use. Since SPARQL queries can be very computationally expensive when the ontology contains high levels of depth, it is more efficient to avoid repeated execution of SPARQL queries by storing previously retrieved information.

Furthermore, although the ontology plays a fundamental role in this

case, the use of SPARQL queries is motivated by the fact that directly providing the row ontology to a language model (as illustrated in Code 5.1, hence an extract containing the various RDF/OWL constructs), could be counterproductive.

Code 5.1: Ontology extract

```
    <owl:equivalentClass>
        <owl:Class>
            <owl:intersectionOf rdf:parseType="Collection">
                <owl:Restriction>
                    <owl:onProperty rdf:resource=#address"/>
                    <owl:someValuesFrom
                        rdf:resource=#CybersecurityForICTProduct"/>
                </owl:Restriction>
                <owl:Restriction>
                    <owl:onProperty rdf:resource=#address"/>
                    <owl:someValuesFrom
                        rdf:resource=#CybersecurityForICTService"/>
                </owl:Restriction>
            </owl:intersectionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource=#Art7Par2-MemberState-Compliant"/>
</owl:Class>
```

The ontological constructs can influence the semantics of the text and potentially alter the meaning of the measures, and may superfluously occupy the length of the context window, a typical parameter of models that establishes how much knowledge a model can handle with one execution. The length of the context window is highly dependent on the language models themselves and how advanced they are. It is a variable factor that can affect the correctness and completeness of the final outputs. If certain information is truncated if it exceeds the set length, then it would make the final output both incorrect and incomplete.

Therefore, once a precise request has been made for the entities to be found (in our case the articles of the Directive in which the measures we want are), the SPARQL query can be designed in two ways: directly search for the entity (the Article) and the relative measures; search for all the possible entities and relations that have been defined in the query and then filter at the end only the trace generated in correspondence with our request

A query extract we employed is illustrated in Code 5.2. Naturally, it is not the only solution to solve the problem (a different version has already

been provided in the analysis of SecOnto), but it shows the handling of nested ontological properties and constructs.

Code 5.2: Query for retrieving information from ontology

```
  SELECT DISTINCT ?class_n ?property ?value WHERE {{
{{
  # Direct properties
  ?class_n ?property ?value .
}} UNION
{{
  # Handling equivalent class intersections
  ?class_n owl:equivalentClass ?equiv .
  ?equiv owl:intersectionOf|owl:unionOf ?list .
  ?list rdf:rest*/rdf:first ?value .
  BIND("intersectionMember" AS ?property)
}} UNION
{{
  # Handling Restrictions (someValuesFrom)
  ?restriction rdf:type owl:Restriction ;
               owl:onProperty ?property ;
               owl:someValuesFrom ?value .
  ?class_n owl:equivalentClass|rdfs:subClassOf ?restriction .
}} UNION
{{
  # Handling nested restrictions (deeper levels)
  ?class_n owl:equivalentClass|rdfs:subClassOf ?nested .
  ?nested owl:intersectionOf|owl:unionOf ?list1 .
  ?list1 rdf:rest*/rdf:first ?item1 .
  {{
    # Level 1 Restrictions
    ...
  }} ... }}
```

After their extraction, the triples are stored in an external file, adopting a template that can be more easily interpreted by the language model in the Final Answer Generation sub-step. The template has the following simple format: *Class:*, *Action:*, *Object:*. So, given the triple of the previous example, we obtain: Class: MemberState, Action: adopt, Object: NationalCybersecurityStrategy. An extract is illustrated in Code 5.3.

Code 5.3: Ontology presented in a *gamified* format for prompt engineering

```
1 Class: Art21Par2-a-Entity-Compliant, Action: include, Object:
      InformationSystemSecurityPolicy
2 Class: Art21Par2-a-Entity-Compliant, Action: include, Object:
      RiskAnalysisPolicy
3 Class: Art21Par2-b-Entity-Compliant, Action: include, Object:
      IncidentHandling
4 Class: Art21Par2-c-Entity-Compliant, Action: include, Object:
      BackupManagement
5 ...
```

The method we employed for retrieving the chosen knowledge from the ontology is required due to technical constraints. Modern language models do not provide libraries capable of directly querying an ontology in OWL/RDF format. Furthermore, our experiments confirmed that providing the ontology directly to the model as textual input is ineffective. In case of a sufficiently large ontology, the model would not be able to parse its content and produce irrelevant output. As previously mentioned, the ontological constructs — recognisable as tags — may influence the interpretation of the model. The model cannot discern the ontological constructs as superfluous for its decision-making process.

Therefore, extracting specific knowledge via SPARQL and storing it for subsequent use is strongly recommended.

5.6.1.3 Attack patterns retrieval

This sub-step consists of retrieving the attack patterns from CAPEC contained in a given file. In this step, we only interact with a CSV file, the CAPEC native format provided by MITRE. However, the file format is not a constraint, provided that the knowledge to be extracted, i.e., the attack patterns, is not intrinsically tied to a specific format, unlike in the previous step.

Rather than interacting with the CSV using the standard libraries, we employed Langchain (a framework for managing AI agents [77]), which is more suited for an AI-oriented application. After extracting the attack patterns with the CSVLoader method, we leveraged the capabilities of the model to retrieve and store each attack pattern and the related fields. We used the standard phases of model interaction: creation, prompt invocation and final response generation to extract the attack patterns and store them with a very precise scheme. This facilitates the understanding of the model in the last step for the correlation phase.

This approach is generalisable, as long as a structured source like CAPEC is still considered. This means the input does not necessarily have to be a CSV file, for example. In fact, unlike the first sub-step, this second step can be implemented directly, thanks to the modern libraries made available to models that can parse different formats of documents.

Using an AI-oriented method for extracting a simple CSV file allows us to accomplish a first round of checks; we can verify if the model employed and the parameters used for instantiating the model are correct and sufficient to handle the requests of Step 3. Furthermore, we can test different models and their understanding capabilities before the correlational step.

5.6.1.4 Final answer generation

This sub-step consists of correlating the security measures from the NIS 2 Directive with the attack patterns from CAPEC.

Before prompting the language model to identify correlations, the method employs the few-shot prompt strategy to further improve and drive the decisional process. The model is provided with a series of examples about the desired output schema and demonstrations of correlations we intend as security analysts. While the approach is ontology-driven, explicit instructions and examples are still required to help the model contextualise the task.

Although we designed the few-shot approach, this sub-step fully relies on the knowledge retrieved in the first and second steps, stored in the internal state of the agent. As illustrated in Code 4, the ontology-driven prompt engineering is implemented by providing the variables *docs_content* and *csv_analysis*. The two variables, respectively, encapsulate the measures obtained from the ontology with the second sub-step and the attack patterns extracted from the CSV file with the third sub-step.

The prompt is not executed n times — where n is the Cartesian product of the number of measures and attack patterns — to individuate potential correlations. Instead, having verified that the context window is large enough, we provide the model with the retrieved contents through the above variables. We also provide the model with the desired output schema, composed of: the ontological class, property, and object; the matched attack pattern ID, matched attack pattern name, and matched attack pattern description. The complete scheme requested in the prompt is illustrated in Code 5.4.

Code 5.4: Prompt engineering by leveraging retrieved information

```
Based on the **retrieved text** and **retrieved
CSV attack descriptions**, extract and directly match
    the **most relevant** attack technique for
    each class, action and object. Do not forget any class, action,
        and object.

    **TEXT SOURCE**:
    {docs_content}

    **CSV ATTACK TECHNIQUES**:
    {csv_analysis}
    ---
    **MATCHING TEMPLATE (strictly follow this format)**:
    - **Class Involved**: "<Class>"
    - **Action Performed**: "<Action>"
    - **Object Affected**: "<Object>"
    - **Matched Attack ID**: "<ID from CSV>"
    - **Matched Attack Name**: "<Name from CSV>"
    - **Matched Attack Description**: "<Description from CSV>"
```

We also chose to set a specific pattern for asking the model to motivate matching it was able to find. The motivation is fundamental to check the capabilities of the model, so if the proposed matching is meaningful, and in the end, to help a security analyst who may not have initially imagined the matching that the model proposed. A possible output is illustrated in Code 5.5.

Code 5.5: Output from prompt engineering

```
Matched Attack Patterns from CAPEC:

- **Class Involved**: "Art21Par2-a-Entity-Compliant"
- **Action Performed**: "include"
- **Object Affected**: "RiskAnalysisPolicy"
- **Matched Attack ID**: "575"
- **Matched Attack Name**: "Account Footprinting"
- **Matched Attack Description**: "An adversary exploits
functionality meant to identify information about the domain
accounts and their permissions on the target system."

**Motivation**:
The action 'include' and the object 'RiskAnalysisPolicy' connect
with Account Footprinting, as analyzing risk policies can reveal
information about user accounts and their permissions, which can
be exploited by an adversary.
```

```
17
18 ---
19
20 - **Class Involved**: "Art21Par2-b-Entity-Compliant"
21 - **Action Performed**: "include"
22 - **Object Affected**: "IncidentHandling"
23 - **Matched Attack ID**: "577"
24 - **Matched Attack Name**: "Owner Footprinting"
25 - **Matched Attack Description**: "An adversary exploits
26 functionality meant to identify information about the primary
27 users on the target system to an authorized user."
28
29 **Motivation**:
30 The action 'include' and the object 'IncidentHandling' relate to
31 Owner Footprinting, as understanding incident handling
32 procedures can help an adversary identify key personnel involved
33 in security incidents.
34 ...
```

5.6.2 Applying the Ontological Semantics Step

To apply this step, we mainly used the model gpt-4o-mini by OpenAI. This model was selected after alternative models, including LLaMA 3.2B and DeepSeek, were tested and proved to be ineffective in performing the designated task. Through prompt engineering we asked to find the correlations between the measures taken in their natural form and the attack patterns, considering 3 possible scenarios, in line with the questionnaires: in the first case we only asked to generally find only correlated pairs, in the second case we asked to find a correlation in the case in which the measure *mitigates* an attack pattern, in the third case we asked to find a correlation in the case in which the measure *prevents* an attack pattern. For each scenario, the prompts were executed 15 times. Only correlations identified consistently across all runs were retained. This consistency is interpreted as an indicator of response reliability.

The employed settings used for the Ontological Semantics step and the number of found correlations between security measures and attack patterns are illustrated in the tables 5.19 and 5.20, respectively, for Article 7 and Article 21. In particular, the column *N.Corr.* shows the Number of Correlations obtained using the specific setting of each row, under the column *Setting* — for the Ontological Semantics step, consider only the setting where the acronym **PE** appears.

5.7 Validation Base Construction Step

The Validation Base Construction step aims to build a *ground truth* for evaluating and validating the semantic methods. As previously noted, no existing ground truth correlates security measures from legal sources to attack patterns, particularly considering the NIS 2 Directive and CAPEC. Consequently, the method we adopted relies on the expertise of security experts. Through carefully designed questionnaires, the experts express their assessments regarding the correlations between security measures and attack patterns.

In the following subsections, we show the process of creation and administration of the questionnaires, as well as how we evaluated and interpreted the collective results, with the final goal of creating a ground truth.

5.7.1 Method for the validation base construction step

As a second step of WISARD, we focus on creating the questionnaires by selecting the main categories of information to be managed therein. We selected three main categories: the security measures, the attack patterns, and the user input categories.

Selection of security measures Since the NIS 2 Directive is quite heterogeneous, we focused our study on two articles, Article 7 and Article 21, and of both, we considered only the second paragraph. The articles and paragraphs have been selected following an *applicability criterion*: unlike the others, they contain technical and technological measures that are more applicable in a security context. Other articles may contain more organisational measures, e.g., regarding the timing of data breach notifications, which therefore may not be linked to attack patterns a priori.

The total number of measures is 20, so we created 20 questionnaires, each relating to a single measure.

Selection of attack patterns The attack patterns were selected from the CAPEC framework, which contains 559 attack patterns in total. We chose only 29 attack patterns based on the criterion of *specificity* and *relevance*. For the specificity criterion, we considered only the attack patterns classified as *meta* — according to CAPEC, attack patterns develop in hierarchical trees, where a tree contains three possible sequential

classification nodes: meta (for abstract characterisation), standard (for specific methodology), detailed (for higher level of detail). This criterion allows us not to have attack patterns that are too detailed and that could already be considered uncorrelated with the security measures. For the relevance criterion, we considered attack patterns with likelihood and severity *medium* and *high*, and all the possible combinations considering these two values. This criterion allows us to have attack patterns that are more likely to occur. In each questionnaire, the attack patterns are listed in the rows.

Selection of user input categories We proposed 4 selectable correlation categories to be filled. Two categories for indicating the relationship that correlates an attack pattern with a measure. To make the correlation more explicit and appropriately reflect the nature of a security measure, the categories are *Prevents* (for indicating that a measure can prevent the attack pattern) and *Mitigates* (for indicating that a measure can mitigate an attack pattern). A third category, *Unsure*, is used to allow experts to express uncertainty between the first two categories. A fourth category, *Do not understand*, is used to allow experts to express a misunderstanding of the possible correlation between the specific measure and specific attack pattern. For each category, the respondent can select a *tick* (✔) for indicating correlation, a *cross* (✘) or a blank space for indicating no correlation. In each questionnaire, the filling categories are listed in the columns.

An extract of the questionnaire, and a possible way of filling it, is shown in Table 5.13.

Table 5.13: Example of a questionnaire

***Measure**: policies and procedures regarding the use of cryptography*

Attack Pattern	Prevents	Mitigates	Unsure	Do not understand
Exploitation of Trusted Identifiers	✔	✘		
Exploiting Trust in Client			✔	
Leveraging Race Conditions	✔			
Fuzzing	✘		✔	
Manipulating State		✔		
Adversary in the Middle (AiTM)	✘	✔		
Interface Manipulation		✔		
...		✔		
Object Injection				✔

5.7.2 Applying the validation base construction step

The questionnaires were submitted to 9 security experts belonging to the University of Twente. Therefore, we collected and categorised the results so that they could be used interoperably and, above all, analysed. The heatmaps in the following figures (5.7, 5.8, 5.9, 5.10) show, in summary, what the distribution of correlations was for each article and each attack pattern. There are four scenarios due to the combinations of the two articles with the two categories of filling *prevents* and *mitigates*.

The figures show, for each attack pattern, the number of correlations found for each article. The figures show how relevant an attack pattern was considered for the prevention and mitigation categories. A higher position in the heatmap indicates that an attack pattern is more considered mitigable or preventable by the measures contained in the single reference article.

Of particular interest, however, is the extent to which these correlations are shared by the various experts. This analysis is essential to understand how much the answers were correlated among the respondents, consistent and consequently useful for defining the concept of ground truth. We calculated two statistical indices, widely used in contexts with multiple raters, Fleiss's kappa and Krippendorff's alpha, and a third index based on the percentage of agreement. The indices were calculated individually for each measure and the prevention- and mitigation-related categories. In the end, an average was calculated to determine the general trend for each article. The obtained values from the indices are illustrated in Table 5.4 for Article 7, and in Table 5.5 for Article 21.

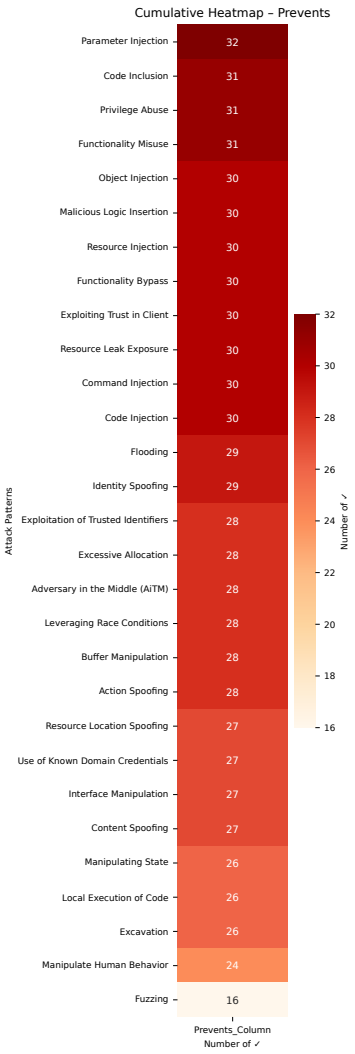


Figure 5.7: Heatmap for response *Prevents* on Article 7

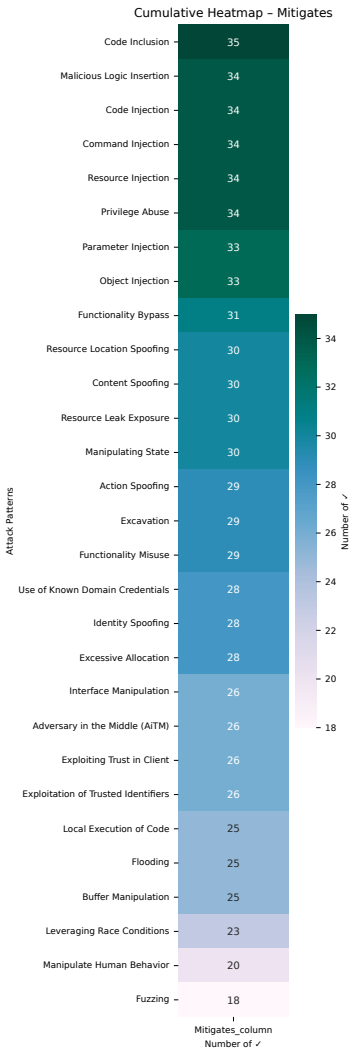


Figure 5.8: Heatmap for response *Mitigates* on Article 7

Table 5.4: Raters Correlation Metrics for Article 7

Measure	Prevents			Mitigates		
	Fleiss	Kripp.	% Acc.	Fleiss	Kripp.	% Acc.
7-2a	-0.09	0.73	0.00%	-0.06	0.64	0.00%
7-2b	-0.09	0.73	0.00%	-0.06	0.54	7.00%
7-2c	-0.09	0.75	3.00%	-0.05	0.57	0.00%
7-2d	-0.07	0.44	24.00%	-0.10	0.79	3.00%
7-2e	-0.09	0.82	0.00%	-0.10	0.84	0.00%
7-2f	-0.05	0.74	0.00%	-0.08	0.75	0.00%
7-2g	-0.12	0.97	0.00%	-0.10	0.85	0.00%
7-2h	-0.10	0.86	0.00%	-0.11	0.85	3.00%
7-2i	-0.12	0.90	0.00%	-0.10	0.80	0.00%

Continuing in the next page

Table 5.4 – Continuation

Measure	Prevents			Mitigates		
	Fleiss	Kripp.	% Acc.	Fleiss	Kripp.	% Acc.
7-2j	-0.11	0.82	0.00%	-0.10	0.80	0.00%
Average	**-0.09**	**0.78**	**3.00%**	**-0.09**	**0.74**	**1.00%**

Table 5.5: Raters Correlation Metrics for Article 21

Measure	Prevents			Mitigates		
	Fleiss	Kripp.	% Acc.	Fleiss	Kripp.	% Acc.
21-2a	-0.09	0.67	0.00%	-0.10	0.74	0.00%
21-2b	1.00 (✘)	1.00 (✘)	100.00% (✘)	-0.07	0.77	0.00%
21-2c	1.00 (✘)	1.00 (✘)	100.00% (✘)	-0.07	0.56	0.00%
21-2d	-0.04	0.42	21.00%	-0.08	0.71	0.00%
21-2e	-0.03	0.40	21.00%	-0.09	0.78	0.00%
21-2f	-0.09	0.76	0.00%	-0.10	0.69	0.00%
21-2g	0.04	0.44	3.00%	-0.02	0.40	24.00%
21-2h	0.37	0.05	45.00%	-0.04	0.26	45.00%
21-2i	0.06	0.09	41.00%	-0.03	0.42	17.00%
21-2j	0.17	0.07	48.00%	0.05	0.39	17.00%
Average	**0.24**	**0.49**	**38.00%**	**-0.06**	**0.57**	**10.00%**

For both articles, we can interpret the indices in the following way:

1. Fleiss's kappa: The low value of Fleiss's kappa indicates inconsistency among the security experts, i.e. they did not agree on the correlations between security measures and attack patterns. The negative values suggest that the agreement among the experts is worse than the agreement expected by chance.

2. Krippendorff's alpha: The high value for Krippendorff's alpha indicates that the experts have consistent individual strategies, following identifiable patterns.

3. Agreement rate: The low agreement rate indicates that the experts had poor full agreement in the measures/attack patterns correlations. If all the respondents provided the same binary value (1 or 0), there is full agreement; otherwise, if at least one respondent answered with a different value, there is no agreement.

Consequently, we can draw the following general conclusions on the Validation Base Construction step:

1. security experts would encounter significant challenges in implementing the measures in place in the directives and, above all, effectively finding possible attacks on the measures found non-compliant. The main cause is the clear difference between the

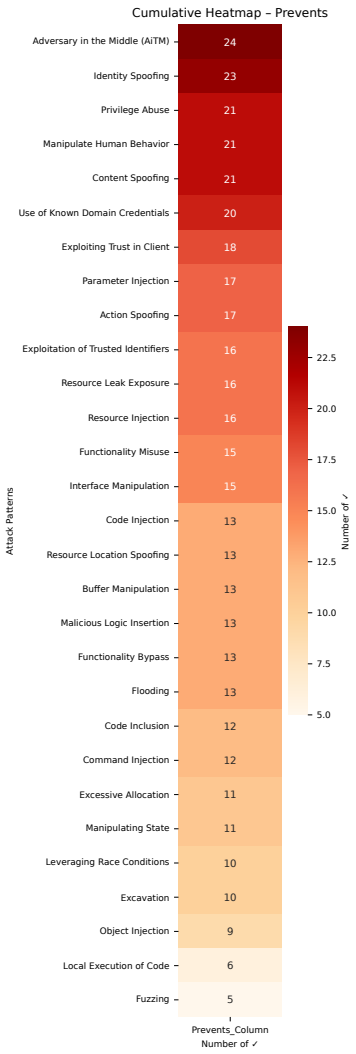


Figure 5.9: Heatmap for response *Prevents* on Article 21

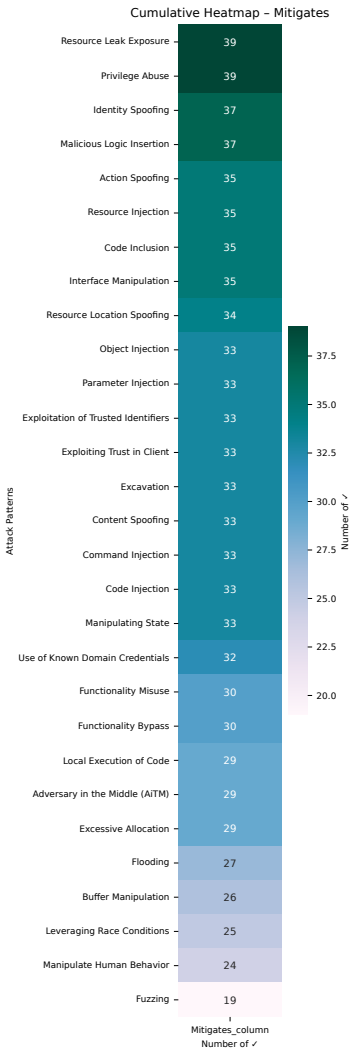


Figure 5.10: Heatmap for response *Mitigates* on Article 21

general and non-specific content of a security Directive and the applied security of the systems and software;

2. the results demonstrate that there is far too much divergence in the collective opinions of experts; therefore, a collective agreement (a ground truth) cannot be properly created.

To still leverage the security experts' responses for validation purposes, the results obtained will not be considered as ground truth, but as a *support element*, hence a validation base, for the validation of the

methods of the Semantic step (Semantic Similarity step and Ontological Semantics step). Since it would be inaccurate to claim that they constitute a ground truth, we will exploit the results of the questionnaires obtained to intersect the results derived from the application of the methods of the Semantic step. Rather than validating the individual semantic methods, this approach allows us to validate the measures/attack patterns correlations. The precise criteria for selecting the correlations from the questionnaires will be discussed in the next section.

5.8 Intersection Step

The final step consists of intersecting the results obtained from the Validation Base Construction step with the Semantic Similarity step and the Ontological Semantics step. By doing so, the uncertainty of security experts is corroborated by the semantic methods; vice versa, the results of the semantic methods are themselves confirmed through the convergence of the response from security experts.

5.8.1 Method for the intersection step

For the final validation and intersection between the human approach (creation of the validation base) and the semantic methods, the method consists of: establishing a *convergence criterion* based on expert agreement; establishing a *reachability criterion*, to consider the largest (but meaningful) number of attack patterns; establishing the *intersection criteria* of the results of the various steps; and evaluating the results obtained.

Convergence criterion A security measure/attack pattern association is considered valid if among the raters (security experts), the sum of all the preferences obtained in the *prevents*, *mitigates*, and *unsure* columns is greater than or equal to half of the raters. If the raters put a tick (✔) in the prevention and mitigation columns, they are respectively worth a point, while if the tick has been inserted in the column of unsure, then it will be worth 0.5. If a cell is left blank or marked with a cross (✘), in both cases the answer will be null.

Considering the answers given by two fictitious experts X and Y, relating to a specific measure, a possible way of filling the questionnaire is illustrated in Table 5.6, which is then evaluated according to the convergence criterion.

Table 5.6: Example of a questionnaire

***Measure**: policies and procedures regarding the use of cryptography*

Rater	Attack Pattern	Prevents	Mitigates	Unsure	Do not understand
X	Exploitation of Trusted Identifiers	✔	✘		
X	Exploiting Trust in Client			✔	
X	Leveraging Race Conditions	✔			
X	Fuzzing			✔	
X	Manipulating State			✔	
X	Adversary in the Middle (AiTM)	✘	✔		
X	Interface Manipulation		✔		
X	...	...	...	...	...
X	Object Injection				✔
Y	Exploitation of Trusted Identifiers	✔	✔		
Y	Exploiting Trust in Client			✔	
Y	Leveraging Race Conditions	✔			
Y	Fuzzing	✔			
Y	Manipulating State				✔
Y	Adversary in the Middle (AiTM)		✔		
Y	Interface Manipulation	✘	✔		
Y	...	...	...	...	...
Y	Object Injection	✘			

Assuming the scores that we specified previously, we obtain the sums in the Table 5.17. The attack patterns highlighted in green are those that are convergent according to the established criteria — the example has two raters, consequently, half is 1. Values greater than or equal to 1 are considered valid according to the convergence criterion.

Table 5.17: Example of Convergence among raters

Attack Pattern	Convergence
Exploitation of Trusted Identifiers	3
Exploiting Trust in Client	1
Leveraging Race Conditions	2
Fuzzing	1.5
Manipulating State	0.5
Adversary in the Middle (AiTM)	2
Interface Manipulation	2
Object Injection	0

Reachability criterion To address the limited number of attack patterns provided within the questionnaires — only 29 attack patterns were included in the questionnaires, chosen with the discussed criteria — we applied our methods to an extension of the attack patterns. In particular, since in the questionnaires we only chose the attack patterns classified as *meta*, we also consider their direct children to extend the set, indicated in CAPEC with the nomenclature as *ChildOf*.

The motivation is that if a correlation exists between a measure and an attack pattern, then it is reasonable to assume a correlation may exist with its descendants. CAPEC provides other types of relations, i.e., *CanPrecede* or *CanFollow*. These relations do not belong to the same lineage tree, but indicate whether, in an attack strategy, the chosen attack pattern can be used before or after another attack pattern — attack pattern 98 (Phishing), according to CAPEC, precedes attack pattern 543 (Counterfeit Websites). However, since they are not attack patterns belonging to the same lineage, and therefore not correlated by nature, we cannot apply the same considerations as in the ChildOf case.

We applied the extension in two cases: **Case A**, both to the attack patterns obtained from the semantic methods, and to the attack patterns obtained from the questionnaires; **Case B**, only to the attack patterns obtained from the questionnaires. Given that the original questionnaire includes a fixed set of 29 attack patterns, their extension across all the experiments is also fixed. From the 29 attack patterns, we reach 136 attack patterns in total; therefore, 107 more attack patterns are descendants of the 29 fixed ones.

We also implemented the extension to offer a degree of flexibility to security professionals who will use the final mapping we present. Instead of providing a rigid and static mapping, which might be less effective due to its lack of adaptability, the proposed extension introduces an element of discretion. This allows security professionals to independently assess and determine the relevance of specific correlations between attack patterns and security measures, thus supporting informed decision-making based on their contextual needs.

Intersection criterion The intersection criterion is quite simple. Will be considered as *validated* the security measure/attack pattern couple that can be obtained from one of the possible semantic methods and/or their combinations (*settings*), and that is at the same time *confirmed* by the convergence criterion.

5.8.2 Applying the intersection step

By applying the semantic methods and considering previous criteria, we obtain the results illustrated in Tables 5.19 and 5.20. The tables contain 7 columns, and each row contains the following items — columns 3, 4, 5, and 6 contain a double value indicating the results of the experiments considering the above-defined **Case A** and **Case B**, respectively.

Setting The setting is the specific semantic method applied. The setting can be: 1) individual semantic similarity (with normal version of measures and extended measures through 3 models), 2) prompt engineering for ontological semantics (with three different types of prompts to emphasise the terms of prevention and mitigation), 3) intersection of semantic similarity settings (1) with ontological semantics settings (2).

Number of Correlations (N.Corr.) The number of security measure/attack pattern correlations obtained by applying the semantic methods, relative to the specific setting shown.

True Positives (TPs) The number of correlations identified based on the relative setting and deemed valid according to the validation base, i.e., recognised as valid from the convergence criterion.

Metrics The metrics Precision (P), Recall (R), and F1 (F1) are used for the related setting. We calculated evaluation metrics to assess the performance of the semantic methods. Their formulation is outlined in Table 5.18 – also useful in the continuation of the dissertation.

Table 5.18: Mathematical representation of the machine learning metrics

Metric	Definition and Formula
True Positives (TP)	Number of positive examples predicted positive: $TP = \sum_{i=1}^{n} \mathbb{1}[y_i = 1 \wedge \hat{y}_i = 1]$
False Positives (FP)	Number of negative examples predicted positive: $FP = \sum_{i=1}^{n} \mathbb{1}[y_i = 0 \wedge \hat{y}_i = 1]$
False Negatives (FN)	Number of positive examples predicted negative: $FN = \sum_{i=1}^{n} \mathbb{1}[y_i = 1 \wedge \hat{y}_i = 0]$
True Negatives (TN)	Number of negative examples predicted negative: $TN = \sum_{i=1}^{n} \mathbb{1}[y_i = 0 \wedge \hat{y}_i = 0]$
Accuracy	$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$
Precision	$\text{Precision} = \frac{TP}{TP + FP}$
Recall	$\text{Recall} = \frac{TP}{TP + FN}$
F1 Score	$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Although the Validation Base Construction step confirmed the absence of an established ground truth for definitive and objective validation, the metrics are designed to provide information on the relative effectiveness and alignment of the semantic methods under consideration.

Table 5.19: Methodology applied on Article 7

Setting	N.Corr.	TPs (A)/TPs(B)	P(A)/P(B)	R(A)/R(B)	F1(A)/F1(B)
SS - AMax (2 models)	11	4/3	0.17/0.14	0.0	0.01
SS - RMax (4 models)	70	30/14	0.26/0.14	0.03/0.01	0.05/0.02
SS - RMax (3 models)	287	84/58	0.19/0.14	0.08/0.05	0.11/0.08
SS - RMax (2 models)	1882	440/312	0.16/0.12	0.40/0.32	0.22/0.17
PE (asking for *Normal*)	45	15	0.12	0.01	0.02
PE (asking for *Prevention*)	42	8/7	0.11	0.01	0.01
PE (asking for *Mitigation*)	47	11/7	0.13/0.09	0.01	0.01
SS (ext. with LLama) - AMax (2 models)	8	6/2	0.46/0.22	0.01/0.0	0.01/0.0
SS (ext. with LLama) - RMax (4 models)	17	3	0.15	0.0	0.01
SS (ext. with LLama) - RMax (3 models)	155	48/27	0.18/0.11	0.04/0.02	0.07/0.04
SS (ext. with LLama) - RMax (2 models)	2037	566/374	0.18/0.13	0.51/0.41	0.27/0.19
SS (ext. with LLama) - AMax (2 models) + PE (Normal)	0	0	0	0	0

SS (ext. with LLama) - RMax (4 models) + PE (Normal)	0	0	0	0	0
SS (ext. with LLama) - RMax (3 models) + PE (Normal)	4	2	0.29	0.0	0.0
SS (ext. with LLama) - RMax (2 models) + PE (Normal)	19	6	0.26	0.01	0.01
SS (ext. with Mistral) - AMax (2 models)	6	3	0.13	0.0	0.01
SS (ext. with Mistral) - RMax (4 models)	74	18/12	0.14/0.1	0.02/0.01	0.03/0.02
SS (ext. with Mistral) - RMax (3 models)	399	112/70	0.18/0.12	0.10/0.07	0.13/0.09
SS (ext. with Mistral) - RMax (2 models)	3128	871/628	0.21/0.16	0.79/0.73	0.33/0.26
SS (ext. with Mistral) - AMax (2 models) + PE (Normal)	0	0	0	0	0
SS (ext. with Mistral) - RMax (4 models) + PE (Normal)	0	0	0	0	0
SS (ext. with Mistral) - RMax (3 models) + PE (Normal)	2	0	0	0	0
SS (ext. with Mistral) - RMax (2 models) + PE (Normal)	29	8	0.13	0.01	0.01
SS (ext. with Phi4) - AMax (2 models)	8	2	0.13	0.0	0.0
SS (ext. with Phi4) - RMax (4 models)	63	10/5	0.10/0.05	0.01/0.0	0.02/0.01
SS (ext. with Phi4) - RMax (3 models)	307	102/63	0.21/0.14	0.09/0.06	0.13/0.08
SS (ext. with Phi4) - RMax (2 models)	2629	579/489	0.16/0.14	0.54/0.49	0.25/0.21
SS (ext. with Phi4) - AMax (2 models) + PE (Normal)	0	0	0	0	0
SS (ext. with Phi4) - RMax (4 models) + PE (Normal)	-	0	0.0	0.0	0.0
SS (ext. with Phi4) - RMax (3 models) + PE (Normal)	3	-	0.33	0.0	0.0
SS (ext. with Phi4) - RMax (2 models) + PE (Normal)	20	7	0.14	0.01	0.01
SS - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS - RMax (4 models) + PE (*Prevention*)	-	0	0	0	0
SS - RMax (3 models) + PE (*Prevention*)	6	0	0	0	0
SS - RMax (2 models) + PE (*Prevention*)	16	-	0.05	0.0	0.00
SS - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS - RMax (4 models) + PE (*Mitigation*)	2	0	0	0	0
SS - RMax (3 models) + PE (*Mitigation*)	9	-	0.11	0.0	0.0

SS - RMax (2 models) + PE (*Mitigation*)	24	2	0.06	0.0	0.0
SS (ext. with LLama) - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS (ext. with LLama) - RMax (4 models) + PE (*Prevention*)	-	0	0	0	0
SS (ext. with LLama) - RMax (3 models) + PE (*Prevention*)	5	2	0.18	0.0	0.0
SS (ext. with LLama) - RMax (2 models) + PE (*Prevention*)	23	2	0.07	0.0	0.0
SS (ext. with LLama) - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with LLama) - RMax (4 models) + PE (*Mitigation*)	-	0	0	0	0
SS (ext. with LLama) - RMax (3 models) + PE (*Mitigation*)	3	-	0.11	0.0	0.0
SS (ext. with LLama) - RMax (2 models) + PE (*Mitigation*)	22	3	0.09	0.0	0.01
SS (ext. with Mistral) - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS (ext. with Mistral) - RMax (4 models) + PE (*Prevention*)	3	-	0.33	0.0	0.0
SS (ext. with Mistral) - RMax (3 models) + PE (*Prevention*)	10	2	0.2	0.0	0.0
SS (ext. with Mistral) - RMax (2 models) + PE (*Prevention*)	32	4/3	0.1/0.08	0.0	0.01
SS (ext. with Mistral) - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with Mistral) - RMax (4 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with Mistral) - RMax (3 models) + PE (*Mitigation*)	5	-	0.0	0.0	0.0
SS (ext. with Mistral) - RMax (2 models) + PE (*Mitigation*)	38	7/3	0.12/0.05	0.01/0.0	0.01/0.01
SS (ext. with Phi4) - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS (ext. with Phi4) - RMax (4 models) + PE (*Prevention*)	3	0	0	0	0
SS (ext. with Phi4) - RMax (3 models) + PE (*Prevention*)	6	2	0.22	0.0	0.0

SS (ext. with Phi4) - RMax (2 models) + PE (*Prevention*)	21	6/5	0.17/0.15	0.01/0.0	0.01
SS (ext. with Phi4) - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with Phi4) - RMax (4 models) + PE (*Mitigation*)	2	0	0	0	0
SS (ext. with Phi4) - RMax (3 models) + PE (*Mitigation*)	4	-	0.25	0.0	0.0
SS (ext. with Phi4) - RMax (2 models) + PE (*Mitigation*)	29	6/5	0.12/0.1	0.01/0.0	0.01

Table 5.20: Methodology applied on Article 21

Setting	N.Corr.	TPs (A)/TPs(B)	P(A)/P(B)	R(A)/R(B)	F1(A)/F1(B)
SS - AMax (2 models)	13	9/3	0.25/0.1	0.01/0.0	0.02/0.01
SS - RMax (4 models)	70	8	0.08	0.01	0.02
SS - RMax (3 models)	392	117/88	0.19/0.15	0.13/0.19	0.16/0.12
SS - RMax (2 models)	2899	619/415	0.15/0.11	0.71/0.62	0.25/0.19
PE (asking for *Normal*)	37	8/4	0.17/0.09	0.01/0.0	0.02/0.01
PE (asking for *Prevention*)	30	21/8	0.4/0.2	0.02/0.01	0.05/0.02
PE (asking for *Mitigation*)	27	17/8	0.37/0.22	0.02/0.01	0.04/0.02
SS - AMax (2 models) + PE (Normal)	0	0	0	0	0
SS - RMax (4 models) + PE (Normal)	4	0	0.0	0.0	0.0
SS - RMax (3 models) + PE (Normal)	8	6/2	0.46/0.22	0.01/0.0	0.01/0.0
SS - RMax (2 models) + PE (Normal)	28	8/4	0.21/0.12	0.01/0.0	0.02/0.01
SS (ext. with LLama) - AMax (2 models)	10	4/3	0.17/0.14	0.0	0.01
SS (ext. with LLama) - RMax (4 models)	60	15/12	0.12/0.1	0.02/0.01	0.03/0.02
SS (ext. with LLama) - RMax (3 models)	261	92/58	0.18/0.12	0.10/0.07	0.13/0.09
SS (ext. with LLama) - RMax (2 models)	3386	654/511	0.14/0.12	0.74/0.70	0.24/0.20
SS (ext. with LLama) - AMax (2 models) + PE (Normal)	0	0	0	0	0
SS (ext. with LLama) - RMax (4 models) + PE (Normal)	-	0	0	0	0
SS (ext. with LLama) - RMax (3 models) + PE (Normal)	2	0	0	0	0
SS (ext. with LLama) - RMax (2 models) + PE (Normal)	20	8/4	0.27/0.15	0.01/0.0	0.02/0.01
SS (ext. with Mistral) - AMax (2 models)	4	7/2	0.33/0.12	0.01/0.0	0.02/0.0

SS (ext. with Mistral) - RMax (4 models)	17	9/4	0.2/0.1	0.01/0.0	0.02/0.01
SS (ext. with Mistral) - RMax (3 models)	158	39/24	0.13/0.09	0.04/0.03	0.07/0.04
SS (ext. with Mistral) - RMax (2 models)	2119	733/426	0.20/0.13	0.83/0.75	0.33/0.22
SS (ext. with Mistral) - AMax (2 models) + PE (Normal)	0	0	0	0	0
SS (ext. with Mistral) - RMax (4 models) + PE (Normal)	0	0	0	0	0
SS (ext. with Mistral) - RMax (3 models) + PE (Normal)	-	0	0	0	0
SS (ext. with Mistral) - RMax (2 models) + PE (Normal)	24	8/4	0.25/0.14	0.01/0.0	0.02/0.01
SS (ext. with Phi4) - AMax (2 models)	13	3	0.09	0.0	0.01
SS (ext. with Phi4) - RMax (4 models)	32	-	0.02	0.0	0.0
SS (ext. with Phi4) - RMax (3 models)	283	38/21	0.08/0.05	0.04/0.05	0.06/0.03
SS (ext. with Phi4) - RMax (2 models)	2469	534/341	0.15/0.10	0.61/0.50	0.24/0.17
SS (ext. with Phi4) - AMax (2 models) + PE (Normal)	0	0	0	0	0
SS (ext. with Phi4) - RMax (4 models) + PE (Normal)	3	0	0.0	0.0	0.0
SS (ext. with Phi4) - RMax (3 models) + PE (Normal)	9	0	0	0	0
SS (ext. with Phi4) - RMax (2 models) + PE (Normal)	34	8/4	0.18/0.1	0.01/0.0	0.02/0.01
SS - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS - RMax (4 models) + PE (*Prevention*)	4	0	0.0	0.0	0.0
SS - RMax (3 models) + PE (*Prevention*)	7	6/2	0.5/0.25	0.01/0.0	0.01/0.0
SS - RMax (2 models) + PE (*Prevention*)	27	20/8	0.43/0.24	0.02/0.01	0.04/0.02
SS - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS - RMax (4 models) + PE (*Mitigation*)	3	0	0.0	0.0	0.0
SS - RMax (3 models) + PE (*Mitigation*)	5	6/2	0.67/0.4	0.01/0.0	0.01/0.0
SS - RMax (2 models) + PE (*Mitigation*)	22	15/7	0.42/0.25	0.02/0.01	0.03/0.02
SS (ext. with LLama) - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS (ext. with LLama) - RMax (4 models) + PE (*Prevention*)	-	0	0	0	0
SS (ext. with LLama) - RMax (3 models) + PE (*Prevention*)	3	0	0	0	0

SS (ext. with LLama) - RMax (2 models) + PE (*Prevention*)	24	21/8	0.45/0.24	0.02/0.01	0.05/0.02
SS (ext. with LLama) - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with LLama) - RMax (4 models) + PE (*Mitigation*)	-	0	0	0	0
SS (ext. with LLama) - RMax (3 models) + PE (*Mitigation*)	2	0	0	0	0
SS (ext. with LLama) - RMax (2 models) + PE (*Mitigation*)	20	17/8	0.27/0.13	0.01/0.0	0.02/0.01
SS (ext. with Mistral) - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS (ext. with Mistral) - RMax (4 models) + PE (*Prevention*)	0	0	0	0	0
SS (ext. with Mistral) - RMax (3 models) + PE (*Prevention*)	-	0	0	0	0
SS (ext. with Mistral) - RMax (2 models) + PE (*Prevention*)	24	20/8	0.50/0.29	0.02/0.01	0.04/0.02
SS (ext. with Mistral) - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with Mistral) - RMax (4 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with Mistral) - RMax (3 models) + PE (*Mitigation*)	-	0	0	0	0
SS (ext. with Mistral) - RMax (2 models) + PE (*Mitigation*)	20	15/7	0.48/0.30	0.02/0.01	0.03/0.02
SS (ext. with Phi4) - AMax (2 models) + PE (*Prevention*)	0	0	0	0	0
SS (ext. with Phi4) - RMax (4 models) + PE (*Prevention*)	3	0	0	0	0
SS (ext. with Phi4) - RMax (3 models) + PE (*Prevention*)	5	0	0.0	0.0	0.0
SS (ext. with Phi4) - RMax (2 models) + PE (*Prevention*)	24	28/6	0.42/0.19	0.02/0.01	0.04/0.01
SS (ext. with Phi4) - AMax (2 models) + PE (*Mitigation*)	0	0	0	0	0
SS (ext. with Phi4) - RMax (4 models) + PE (*Mitigation*)	2	0	0	0	0

SS (ext. with Phi4) - RMax (3 models) + PE (*Mitigation*)	4	0	0.0	0.0	0.0
SS (ext. with Phi4) - RMax (2 models) + PE (*Mitigation*)	21	13/5	0.37/0.19	0.01	0.03/0.01

5.9 Analysis Of WISARD

The totality of correlations and the code implementation of all experiments are publicly available online [22].

In the following subsections, we will show 5 types of analysis we conducted on the obtained results: total and relative calculation of the correlations found; measures with the highest frequency; attack patterns with the highest frequency; attack patterns/measures correlations with the highest absolute frequency; and comparison between semantic similarity settings. The analysis considers 4 scenarios, as the combination of the two articles (Article 7 and Article 21) with *Case A* and *Case B* (§5.8.1). Each article has 10 security measures, each of which is identified in the range [0-9].

5.9.1 Total correlations

The total number of correlations found, also named true positive in the previous tables, is illustrated in the figures 5.11–5.14.

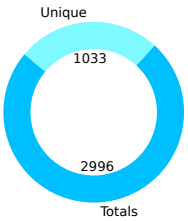


Figure 5.11: Number of correlations of Article 7 - Case A

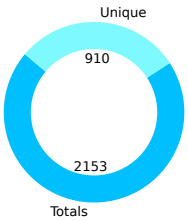


Figure 5.12: Number of correlations of Article 7 - Case B

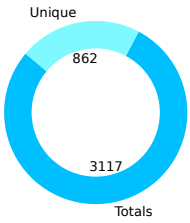


Figure 5.13: Number of correlations of Article 21 - Case A

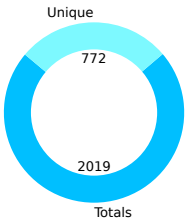


Figure 5.14: Number of correlations of Article 7 - Case B

The distribution of total correlations between the two articles and the two cases is quite consistent. There would be approximately three thousand total correlations found by adding all the results from all the settings. For *Case A* we have: 2996 correlations for Article 7, 3117 correlations for Article 21. While, the transition to *Case B* causes the loss of approximately a thousand correlations: 2154 for Article 7 (843 less), 2019 for Article 21 (1098 less).

Of course, some correlations may have been found from multiple settings. The unique correlations found across all settings are respectively 34.4%, 42,2%, 27.6%, 38.23% with respect to the total correlations found.

Considering all the measures deriving from Articles 7 and 21 (20 measures in total) and all the attack patterns taxonomised in CAPEC (559 Attack Patterns), we would have 11.180 possible correlations. The respective found unique correlations therefore represent respectively the 9.2%, 8.1%, 7.7%, 6.9% of the possible correlations.

5.9.2 Most representative measures

The most representative measures, so the measures that have been mostly correlated with the attack patterns, are illustrated in the figures 5.15–5.18.

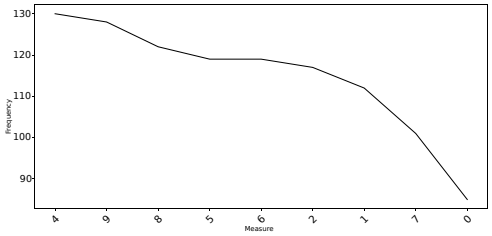


Figure 5.15: Most representative measures of Article 7 - Case A

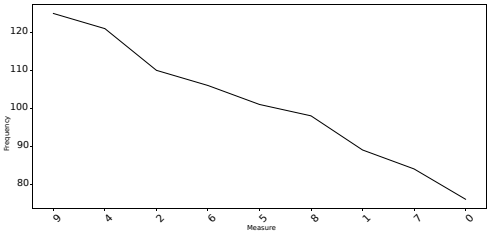


Figure 5.16: Most representative measures of Article 7 - Case B

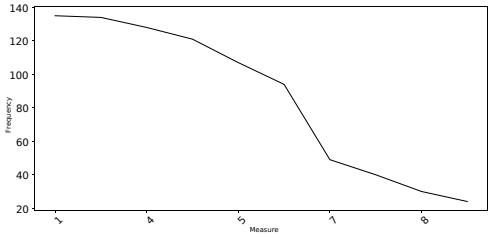


Figure 5.17: Most representative measures of Article 21 - Case A

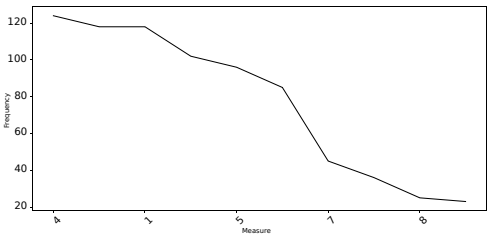


Figure 5.18: Most representative measures of Article 21 - Case B

The most representative measure among the four scenarios is measure 4, for both articles 7 (*promoting the development and integration of relevant advanced technologies aiming to implement state-of-the-art cybersecurity risk-management measures* and 21 (*security in network and information systems acquisition, development and maintenance, including vulnerability handling and disclosure*). The measure is present with more than 100 occurrences in the correlations found. Measures 0 and 8, respectively belonging to articles 7 and 21, are the least representative. The graphs also show how the minimum values of the number of occurrences are around one hundred for Article 7, while around twenty for Article 21. In any case, however, no more than 120-130 occurrences are reached for a single measure.

5.9.3 Most representative attack patterns

The most representative attack patterns, so the attack patterns that have been mostly correlated with the security measures, are illustrated in the figures 5.19–5.22.

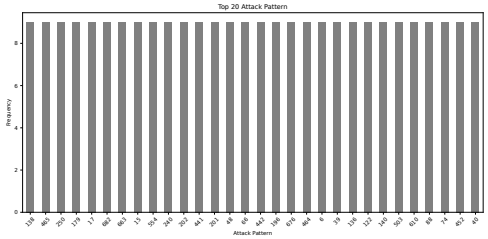


Figure 5.19: Most correlated attack patterns with Article 7 - Case A

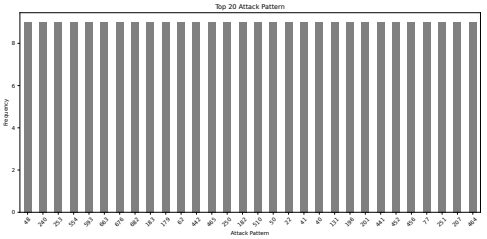


Figure 5.20: Most correlated attack patterns with Article 7 - Case B

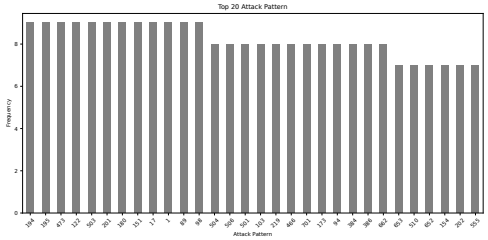


Figure 5.21: Most correlated attack patterns with Article 21 - Case A

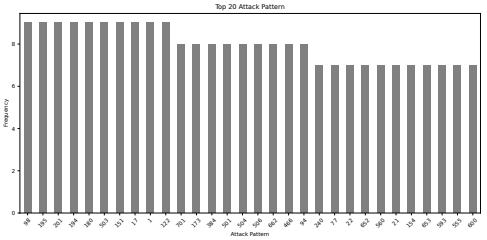


Figure 5.22: Most correlated attack patterns with Article 21 - Case B

In all 4 scenarios considered, the 20 most representative attack patterns occur with at most 9 occurrences. There is an equal distribution of attack patterns, for which there is no attack pattern occurring in more than 9 correlations. Naturally, this result is the consequence of the fact that each article has at most 10 measures, so each attack pattern considered among the most representative (with greater frequency) has been correlated to almost all the measures (9/10). The main difference we can notice is how, in Article 7, the 20 most representative attack patterns have the same frequency of 9 correlations each. Instead, in Article 21, this frequency gradually decreases, with a minimum frequency value equal to 7.

5.9.4 Most representative correlations

The most representative correlations, so the couples attack pattern/security measures that have been mostly found, are illustrated in the figures 5.23–5.26.

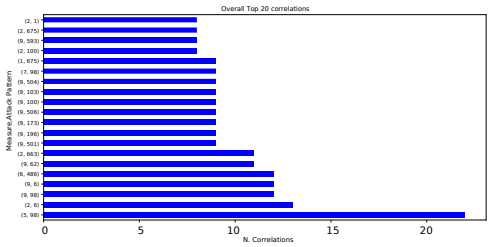


Figure 5.23: Most representative correlations with Article 7 - Case A

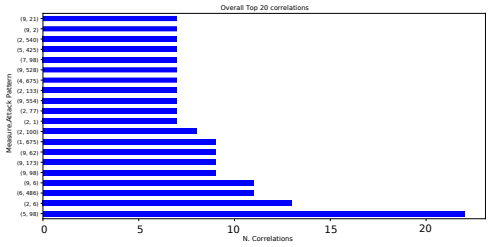


Figure 5.24: Most representative correlations with Article 7 - Case B

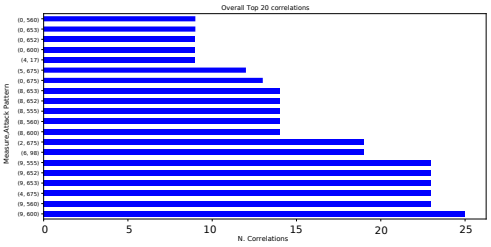


Figure 5.25: Most representative correlations with Article 21 - Case A

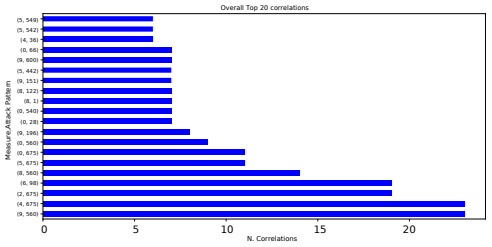


Figure 5.26: Most representative correlations with Article 21 - Case B

These graphs show how often the most representative correlations are produced in output by the various settings. In other words, they are the correlations in which the various settings have converged the most. The most representative correlation is given by measure 9, belonging to Article 21, and attack pattern 600. Which respectively are: *the use of multi-factor authentication or continuous authentication, solutions, secured voice, video and text communications and secured emergency communication systems within the entity*, and Credential Stuffing. Instead, for Article 7, the correlation obtained most frequently is given by the measure 5 (*promoting and developing education and training on cybersecurity, cybersecurity skills, awareness raising and research and development initiatives, as well as guidance on good cyber hygiene practices and controls, aimed at citizens, stakeholders and entities*) and attack pattern 98 (Phishing).

It may be argued that the totality of settings does not reach, for example, 25, which is the highest absolute frequency obtained from a correlation. This quantity is due to an intrinsic intersection existing between some settings. It may be taken, in particular, the case of semantic similarity applied leveraging ***X models***. Of course, applying the same method to a lower quantity, ***X-1 models***, the results of the first application are a subset of the results of the second application, having a looser and less restrictive condition. Of course, intersections could also occur between different settings. For these reasons, the final mapping we present is composed only of unique correlations.

5.9.5 Semantic similarity comparison

This last subsection analyses the trend of the results obtained from semantic similarity. It is of particular interest to us because we have applied the semantic similarity considering the simple measure scenario, and the extended measures scenario with 3 different models, therefore producing 3 new extended measures.

This analysis reveals the extent of overlap between the various scenarios. Specifically, the following tables (Table 5.22 and Table 5.23) compare the intersections between the semantic similarity settings using a specified number of models applied to the standard measure and the corresponding settings applied to the three semantic similarity variants using extended measures with the same set of models. Table 5.21 of acronyms provides a simplified reference to facilitate the interpretation of these intersections.

Table 5.21: Acronyms of the semantic similarity settings

Acronym	Setting
A1	SS - AMax1 (2 models)
A2	SS - RMax2 (4 models)
A3	SS - RMax3 (3 models)
A4	SS - RMax3 (2 models)
B1	SS (ext. with LLama) - AMax1 (2 models)
B2	SS (ext. with LLama) - RMax2 (4 models)
B3	SS (ext. with LLama) - RMax3 (3 models)
B4	SS (ext. with LLama) - RMax3 (2 models)
C1	SS (ext. with Mistral) - AMax1 (2 models)
C2	SS (ext. with Mistral) - RMax2 (4 models)
C3	SS (ext. with Mistral) - RMax3 (3 models)
C4	SS (ext. with Mistral) - RMax3 (2 models)
D1	SS (ext. with Phi4) - AMax1 (2 models)
D2	SS (ext. with Phi4) - RMax2 (4 models)
D3	SS (ext. with Phi4) - RMax3 (3 models)
D4	SS (ext. with Phi4) - RMax3 (2 models)

Table 5.22: Comparison of the semantic similarity settings on Article 7

	B1	B2	B3	B4	C1	C2	C3	C4	D1	D2	D3	D4
A1	0	-	-	-	0	-	-	-	0	-	-	-
A2	-	4	-	-	-	10	-	-	-	18	-	-
A3	-	-	21	-	-	-	104	-	-	-	131	-
A4	-	-	-	776	-	-	-	1192	-	-	-	1118

Table 5.23: Comparison of the semantic similarity settings on Article 21

	B1	B2	B3	B4	C1	C2	C3	C4	D1	D2	D3	D4
A1	0	-	-	-	0	-	-	-	0	-	-	-
A2	-	3	-	-	-	0	-	-	-	7	-	-
A3	-	-	71	-	-	-	43	-	-	-	79	-
A4	-	-	-	1876	-	-	-	1255	-	-	-	1512

The number of intersections found between the various settings consistently follows the rigidity of each semantic similarity algorithm. Since the AMax algorithm produces few correlations due to its absolute maximum condition, the number of intersections is also affected, even being zero in all cases. Instead, although the RMax condition can still be considered restrictive, a good number of intersections are obtained. It can be noted that there is a greater number of intersections found in the case of Article 21 (394 units of difference, considering RMax with 2 models on Phi4). A trend that can already be visualised, starting from the case of RMax with 3 models. In fact, in the case of RMax with 4 models, there is a greater amount of intersection in Article 7, albeit minimal (1, 10, 11, for the respective models).

5.10 Assessing The CWEs

The process of mapping the obtained attack patterns to their associated CWEs is straightforward, as the CAPEC catalogue explicitly includes CWE references for each pattern, requiring no additional inference or correlation. Each CAPEC entry lists its related CWE identifiers in a dedicated field. An attack pattern may reference none, one, or multiple CWEs, depending on whether it targets known structural weaknesses. Conversely, a single CWE may be linked to multiple attack patterns, resulting in a many-to-many relationship between the two taxonomies.

This explicit linkage is by design: CAPEC aims to bridge abstract attack strategies with the software weaknesses they exploit.

Figure 5.27 illustrates an example of a hierarchical relationship between CAPEC, CWE, and CVE. In this example, the attack pattern `CAPEC-112` (Brute Force Attack) exploits the weakness `CWE-521`

(Weak Password Requirements), which can be instantiated in practice as the vulnerability CVE-2020-4574.

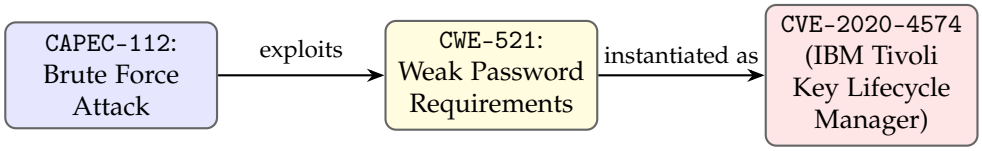


Figure 5.27: Hierarchical relationship between attack patterns (CAPEC), weaknesses (CWE), and specific vulnerabilities (CVE).

In the final chapter of this thesis, we will present a methodology that uses a combination of machine learning and fuzzing to assess software vulnerabilities, hence CVEs, starting from CWEs.

5.11 Automating WISARD

The automation of WISARD can be deconstructed into the automation of the different steps. Regarding the general semantic step, the derivation of attacks can be considered automatic. Both the semantic similarity step and the ontological semantics step employ two different automatic methods: the families of algorithms and prompt engineering, respectively. Therefore, obtaining attack patterns from the security measures is fully automatic, a part not designing the two approaches. However, for both cases, there is little manual component played, respectively, by: the extraction of the security measures Verbatim from the Directive, and the creation of the ontology for prompt engineering. As a consequence, both scenarios strictly refer to the automation of GTCheck and NIS2Onto (hence SecOnto).

The validation base construction step must necessarily be manual. As regards the challenge of creating a ground truth by considering two different security domains, it is a task that cannot be considered to be done automatically. Initially, it requests the creation of curated and detailed questionnaires that simplify and guide the understanding of security experts – hence a socio-technical task – at best. Finally, as we already illustrated when calculating the correlational indexes, even among security experts, a lot of disagreement exists. The disagreement "forced" us to establish different convergence criteria to fully re-employ the correlations provided by the security experts. Therefore, even trying to automate the approach, it would be difficult to trust its reliability.

In the end, by not considering the choice of the different settings

employed for the application of the semantic step, the intersection step can be considered automatic. It simply cross-checks the outcomes of the semantic similarity step and the validation base construction step.

5.12 Concluding Remarks

The work presented in this chapter addresses the promulgation of security directives and the consequent compliance issues. Being compliant means adhering to the security measures defined in directives, but the generality of these measures may be too broad to be easily comparable with the more technical aspects of security. This creates a conceptual gap, particularly concerning the possible attacks that could result from non-compliance with the same measures.

This chapter presents WISARD, a methodology that combines human and artificial intelligence capabilities. WISARD proposes three steps, applied to the security measures of the NIS 2 Directive and the attack patterns of the CAPEC framework. WISARD is composed of two methods for searching correlations between measures and attack patterns, namely *semantic methods*. The concept of semantics is particularly relevant because, to reduce the gap considered, the most meaningful way is to refer to their intrinsic meaning. The first method uses an ensemble of models to evaluate the semantic similarity among the security measures and the attack patterns; the second uses ontologies and prompt engineering to obtain correlations well aligned to the ontological structure of security measures.

To validate the semantic methods, WISARD assesses the existence of correlations between the measures and the attack patterns with the scrutiny of security experts, employed for building the ground truth by completing ad-hoc questionnaires. However, upon calculating the most known correlation indices, we observed significant subjectivity and lack of agreement in responses, which entail two problems: in qualitative terms, we cannot consider the answers as a clear and stable ground truth; in quantitative terms, the answers provided may not reflect the entirety of the correlations, causing the one-to-one evaluation unreliable. Thus, instead of forcing the concept of ground truth, from the answers obtained by the security experts, we designed a validation base by establishing several convergence criteria derived from the experts' responses.

Applying the semantic methods with different settings and language models, and intersecting the semantic methods, produced correlations that were subsequently intersected with the validation base. The final

result is a knowledge base containing the most relevant attack patterns from CAPEC to the security measures from the NIS 2 Directive, which could be practically applied. Upon considering additional attack patterns derived from the attack patterns in the questionnaires, we also have a broader set of correlations. The use of the semantic methods plays a prominent role, without directly involving security experts.

Chapter 6

MOSKAD: From Compliance Gaps within Security Directives to Offensive Killchains

If you know the enemy and know yourself, you need not fear the result of a hundred battles.

— Sun Tzu

This chapter is based on the publication "*Modelling Offensive Security Killchains from Compliance Gaps with Security Directives*" [25].

This chapter illustrates MOSKAD, a methodology for building killchains by combining the attack patterns derived from WISARD into killchains targeting the security measures to which a company may be found to be non-compliant. To create the most accurate kill chain possible, MOSKAD leverages the ATT&CK framework as an intermediate node to correlate attack patterns with any kill chain, by analysing different machine learning techniques to determine the method that most effectively captures these correlations accurately.

6.1 Introduction

Even with proper security measures in place, a change from compliance to non-compliance status may still happen. A company not compliant with the measures contained in such directives may a) become vulnerable to targeted and structured attacks; b) simulate attacks to assess potential weaknesses due to the compliance gaps.

In the previous chapter, we illustrated in detail how to derive specific attack patterns for each security measure analysed. However, these attack patterns must be operationalised — that is, placed within a practical context in which they can be actively applied. In response to the missed security measures, the related attack patterns may be operationalised by structuring them in ordered sequences.

In the security field, an ordered sequence of actions is called a *killchain* [97]. A killchain typically involves a sequence of actions, such as

reconnaissance to identify system vulnerabilities, exploitation of these vulnerabilities, lateral movement through the network and, finally, exfiltration of information. Executing a killchain means executing targeted actions corresponding to specific steps defined within the killchain.

To structure attack patterns into killchains, we designed the methodology MOSKAD (Modelling Offensive Security Killchains from Auditing Directives), composed of three mappings. The first mapping between CAPEC and ATT&CK frameworks, the second mapping between ATT&CK and the steps of any killchain, and the third mapping combines the first two. The first mapping connects CAPEC attack patterns with ATT&CK techniques. The second one is used to find the steps of any killchain starting from the techniques defined in ATT&CK. In the end, a single flow is obtained within the third mapping by combining the first two steps, answering the RQ.

We adopt the ATT&CK framework as a central node for the first two mappings due to two key considerations. First, several existing studies have been presented on mapping ATT&CK with other security frameworks, such as CAPEC, allowing us to evaluate the methods we test for the first mapping. Second, the structure of ATT&CK, divided into techniques and tactics (or *killchain steps*), is itself a killchain, i.e. it can be considered as a ground truth for validating the second mapping. A technique describes a specific action used by an attacker, e.g. phishing, rootkit, etc. A tactic describes the specific goal an attacker aims to accomplish, such as initial access to systems or privilege escalation. To evaluate the mappings, we test three approaches: supervised learning, zero-shot classification, and semantic similarity. Although there are technical intersections (e.g., the use of transformer-based encoders), these approaches differ in their assumptions, data requirements, and inference mechanisms. This comparative evaluation allows us to assess the *best correlational method*, i.e. the method that performs better, and which can be definitively used to capture the proposed mappings. Preliminary experiments reveal that semantic similarity significantly outperforms the other approaches. As a result, we narrow our focus and define the *best correlational method* as the application of semantic similarity across various models.

The software implementation of MOSKAD is openly accessible online [23].

6.2 Related Work

Kwon et al. [76] proposed a cyber threat dictionary by leveraging a mapping between ATT&CK and the NIST Cybersecurity framework. Their work is based on the concept of similarity and clustering to derive similar terms and attributes, and then aggregate attacks to mitigations based on the similarity elements, respectively. The work is based on purely manual mapping. Noor et al. [98] employ the concept of semantics for several steps in a flow that starts with the identification of Tactics, Techniques, and Procedures (TTPs) in order to map and categorise them into Cyber Threat Intelligence Reports (CTIRs) and finally into (ThreatTTP-Detection) TTDs, using ATT&CK techniques. They do not use cosine similarity but rather latent semantic indexing (LSI). Kuppa et al. [75] use cosine similarity as part of a more complex design to map CVEs to ATT&CK techniques automatically. Given the specificity and high level of detail, CVEs are semantically enriched so that they can be more easily linked to known attacks and defences. Straub [127], similarly to our work, discusses the complementarity and use of different killchains, including the ATT&CK killchain and the Lockheed-Martin cyber killchain. The killchain steps are considered conceptually similar to building a framework that combines the steps. The work, therefore, does not consider using one killchain as a pivot to get to others, as well as lacks flexibility due to manual associations. Hemberg et al. [62, 63] have developed the BRON ontology, with which we evaluated the first mapping in this chapter. BRON is a relational graph representing the entries of its various information sources in the field of security, including ATT&CK and CAPEC. Although BRON is an excellent source, it is unclear how the mappings between the various sources were constructed, leaving us to assume that they were done manually or, at the very least, not using semantic similarity as we proposed. From this, it follows that some links may have been lost, especially considering the large number of ATT&CK and CAPEC. Tsuchida et al. [133] propose a method for automatically detecting the relationship between two attack patterns. They fine-tuned a BERT model with the ATT&CK techniques, and like us, they found that although good performance is achieved on classification tasks, worse results are obtained on inference tasks in CAPEC.

Almost all the works discussed here have been derived from a very recent survey chapter [10] where the application of ATT&CK in different scenarios is mainly discussed. ATT&CK, therefore, has been widely used in the state of the art as a central pivot to be able to connect different

security domains. In a combined manner, these works use ATT&CK and, in some cases, the concept of semantic similarity to connect different domains, such as CVE, security reports, and many others, considering textual inputs, among others, of an unstructured nature. However, the contribution we present, of using ATT&CK as a pivot for killchain steps and testing modern semantic similarity models, has not been discussed yet.

6.3 Special Focus On ATT&CK Framework And Killchains

ATT&CK Developed by MITRE, the ATT&CK (Adversarial tactics, techniques, and Common Knowledge) framework [92] is a cornerstone in security. ATT&CK is a global and accessible knowledge base of adversary tactics and techniques based on real-world attacks. ATT&CK is organised into two key components: tactics and techniques. The tactics represent the motivation behind the actions of an adversary and essentially are high-level objects; the tactics span from "Initial Access" to "Execution" and "Exfiltration", among others. The techniques describe how to perform those actions, i.e. the specific methods that an adversary employs to achieve a particular technique; the techniques are numerous, and span from "Phishing", to "PowerShell", and "Data Staging", among others. The techniques may be more granularly subdivided into sub-methods called Sub-techniques. Each description of technique and sub-technique may be further enriched with several details about adversary groups that use them, relevant software and tools, detection methods, and mitigation strategies (upon considering the complementarity with D3FEND).

ATT&CK is employed across numerous security contexts. For threat intelligence analysts, Security Operations Centers (SOCs), red teaming, etc. ATT&CK provides a standardised language for reporting on adversary behaviour, to map their defensive capabilities against known adversary techniques.

Killchains Killchains are ordered sequences of actions that establish the steps to complete a specific task. The term was originally born in the military field but has evolved in the field of cybersecurity, where the task is an *attack*. The killchains, thanks to their generality, can be used both by cyber criminals and by companies specialised in penetration testing activities. Typically, actions in a killchain range from scanning systems to

check for weak access points, to exploiting vulnerabilities for movement within a system, to incremental escalation of privileges to gain control of a system and possibly exfiltrate confidential information.

The introduction of the concept of killchain may lead to misinterpretation due to cases of homonymy in security terminology; the notion of killchain is different from the notion of attack, as they serve different roles in both theoretical modelling and practical threat analysis. While we have asserted that the execution of large-scale attacks can be modelled as killchains, it is important to note that several security taxonomies, such as CAPEC, introduce the concept of an *attack pattern* — a clear case of homonymy. In general, an attack pattern does not share the same structure as a killchain; it does not span multiple steps, but rather represents a specific action or technique applicable within a single step of a broader attack, i.e. a killchain. While the killchain outlines the broader strategic trajectory followed by an adversary, an attack pattern, similarly to an ATT&CK technique, corresponds to a particular method used in one of those steps. A complete attack scenario may therefore consist of multiple attack patterns distributed across the sequential steps of a killchain. A relational model for better emphasising this conceptual difference is illustrated in Figure 6.1.

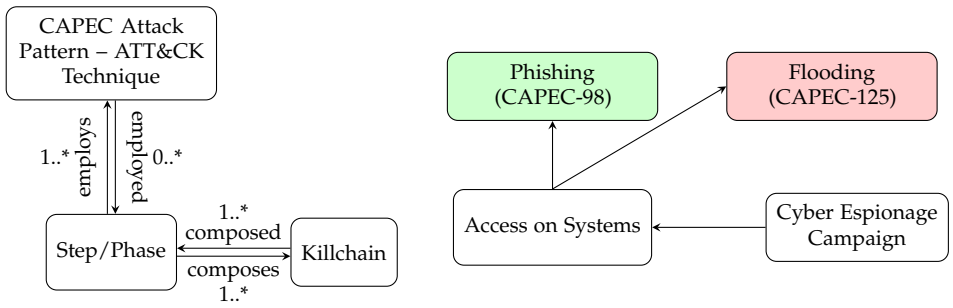


Figure 6.1: Relative model of the killchain concept in relation to methods of attack (left) and its possible instantiation (right)

The relational model exemplifies the relationship that can exist between an attack technique or method (which, for simplicity, can be either a CAPEC attack pattern or an ATT&CK technique) with a structured attack (killchain), composed of steps. An attack method is a necessary element to conduct a killchain step, but not all attack methods, vice versa, can be used for a specific step. For instance, if one wanted to conduct a large-scale, structured cyber espionage attack, one possible step could be to access systems. To access systems, some attack patterns can be used,

but not all. While phishing may be a good candidate, flooding certainly is not, because it is an activity that would otherwise prevent access to systems.

Over time, several killchains have been proposed in the security field, both from academic and industrial sources [97]. Although the ultimate goal is always to attack a system, killchains can be very different from each other and present different actions (*steps*) that vary both quantitatively and qualitatively. For the analysis presented in this chapter, we used 3 killchains: ATT&CK [92], a security framework proposed by MITRE, which, in addition to providing attacks, also provides killchain steps, called tactics; the Lockheed-Martin Cyber killchain [84]; and the Unified Killchain [106]. The steps of each are shown in Table 6.1.

Table 6.1: Illustration of the steps of the main killchains

killchain	Steps
MITRE ATT&CK	(1) Reconnaissance, (2) Resource Development, (3) Initial Access, (4) Execution, (5) Persistence, (6) Privilege Escalation, (7) Defense Evasion, (8) Credential Access, (9) Discovery, (10) Lateral Movement, (11) Collection, (12) Command and Control, (13) Exfiltration, (14) Impact
Cyber killchain (Lockheed-Martin)	(1) Reconnaissance, (2) Weaponization, (3) Delivery, (4) Exploitation, (5) Installation, (6) Command & Control, (7) Actions on Objective
Unified killchain	(1) Reconnaissance, (2) Resource Development, (3) Delivery, (4) Social Engineering, (5) Exploitation, (6) Persistence, (7) Defense Evasion, (8) Command and Control, (9) Pivoting, (10) Discovery, (11) Privilege Escalation, (12) Execution, (13) Credential Access, (14) Lateral Movement, (15) Collection, (16) Exfiltration, (17) Impact, (18) Objectives

Aside from the identical steps among the various killchains, some are highly semantically correlated without empirical proof. For example, the Installation and Exploitation steps in Lockheed-Martin are *semantically similar* to the Persistence and Initial Access steps in ATT&CK. At the same time, the Delivery/Social Engineering and Pivoting steps in the Unified killchain are semantically similar to the Initial Access and Lateral Movement steps in ATT&CK.

As we discuss in more detail in the next section, these killchains have a different role. Briefly, the ATT&CK killchain acts as a pivot between attacks and the steps of the other killchains.

6.4 The MOSKAD Methodology

Before discussing in detail each of the MOSKAD mappings, in this Section we briefly describe the rationale behind the mappings and the main techniques employed in MOSKAD. As previously introduced, MOSKAD is composed of three mappings, linked together by the workflow illustrated in Figure 6.2.

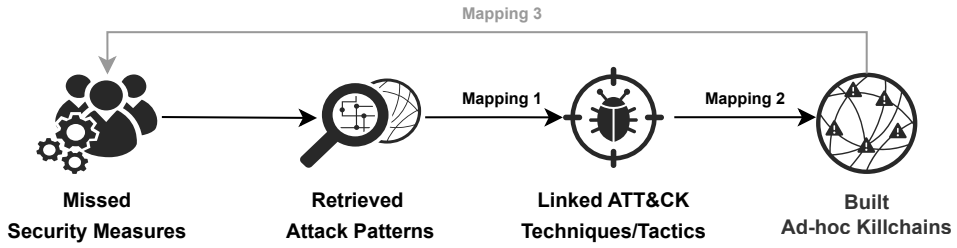


Figure 6.2: MOSKAD overview

Mapping 1: CAPEC to ATT&CK In this mapping, we illustrate how, from a set of techniques established in the ATT&CK framework, we can extract a list of semantically-aligned attack patterns from the CAPEC framework. We validate the correctness of the correlations using a twofold approach: expert validation and cross-referencing with the BRON ontology [62].

Mapping 2: ATT&CK to killchains In this mapping, we show how to obtain the steps of various killchain starting from ATT&CK tactics. The division of ATT&CK into techniques and tactics makes ATT&CK both a killchain (*tactics* are the *steps* of any killchain) and a ground truth on which to test different methods and evaluate the performance of each one.

Mapping 3: Killchains to security measures Finally, we show the integration of the results of the first two mappings to construct ad-hoc killchains corresponding to security measures missed by an organisation. This integration allows for structured killchains, where the attack methods are the attack patterns from CAPEC derived by the application of WISARD.

For each step that requires a mapping to be experimented, hence a correlation, we tested several methods for finding what we call the *best correlational method*, i.e. the method that most accurately find correlations

between CAPEC attack patterns and ATT&CK techniques (mapping 1), and correlations between ATT&CK techniques and the steps of any killchain (mapping 2).

For both mappings, we primarily employed semantic similarity computed via cosine similarity across various embedding models. We tested multiple models for identifying the model that individually performs best in correlating the inputs of each mapping. Additionally, for the second mapping, we explored K-Means clustering. This method was used to map similar techniques to predefined classes, i.e. killchain steps, and to test how much it differs in terms of performance concerning semantic similarity. For the semantic similarity, we employed 4 models: all-Minilm-L6-v2 (Model 1), all-mpnet-base-v2 (Model 2), paraphrase-multilingual-mpnet-base-v2 (Model 3), and attack-BERT (Model 4). All-MiniLM-L6-v2 and all-mpnet-base-v2 are among the best-performing general models according to the STS benchmarks; paraphrase-multilingual-mpnet-base-v2 extends similarity modelling to multilingual settings; attack-BERT is a domain-adapted model trained on cyber threat intelligence data.

The semantic similarity is prioritised due to *conceptual* and *experimental* motivations. The first motivation is *conceptual*: there is a conceptual alignment between the sources used in mappings 1 and 2 (CAPEC and ATT&CK), making semantic similarity not only more computationally efficient but also better suited for capturing these conceptual relationships. The second motivation is *experimental*: in preliminary tests, we tested traditional machine learning techniques, which did not yield satisfactory results.

6.4.1 Preliminary Test #1

In the first preliminary test, we fine-tuned on the ATT&CK framework a pre-trained model to evaluate its performance on CAPEC in an out-of-sample setting, as we employ ATT&CK in both mappings 1 and 2.

In the first case, we fine-tuned the pre-trained transformer-based model roberta-base (an evolution of BERT, and whose performance is proven to be good on generalisation [133]) and tested it over 10 training epochs. Fine-tuning leads to quite good performance metrics when applied for classification tasks on ATT&CK itself (Figure 6.3).

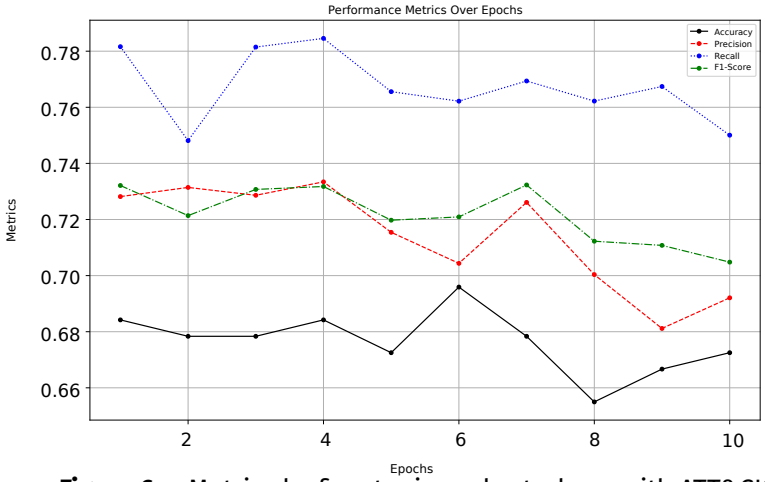


Figure 6.3: Metrics by fine-tuning roberta-base with ATT&CK

However, when the same model is used for the inference task on the attack patterns from CAPEC, a notable drop in accuracy can be observed. Despite the conceptual similarities between attack patterns and ATT&CK techniques, the model struggles to generalise. When comparing the inferences of the models with semantic similarity, we can observe that, when modifying the various semantic similarity models previously stated, there is a significant discrepancy between the attack patterns the model infers and those identified as semantically closest using the embedding models discussed previously. As illustrated in Figures 6.4, 6.5, 6.6, 6.7, this gap persists across all tested embedding models.

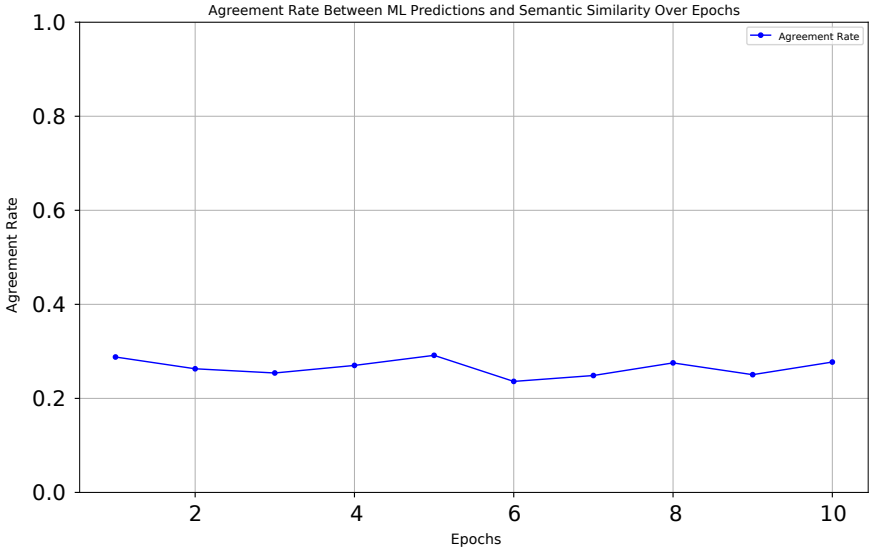


Figure 6.4: Agreement rate with Model 1

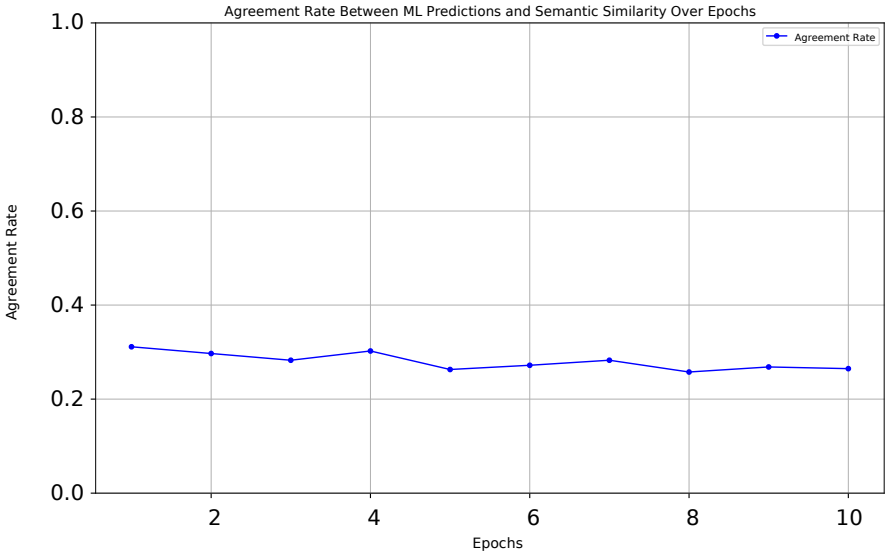


Figure 6.5: Agreement rate with Model 2

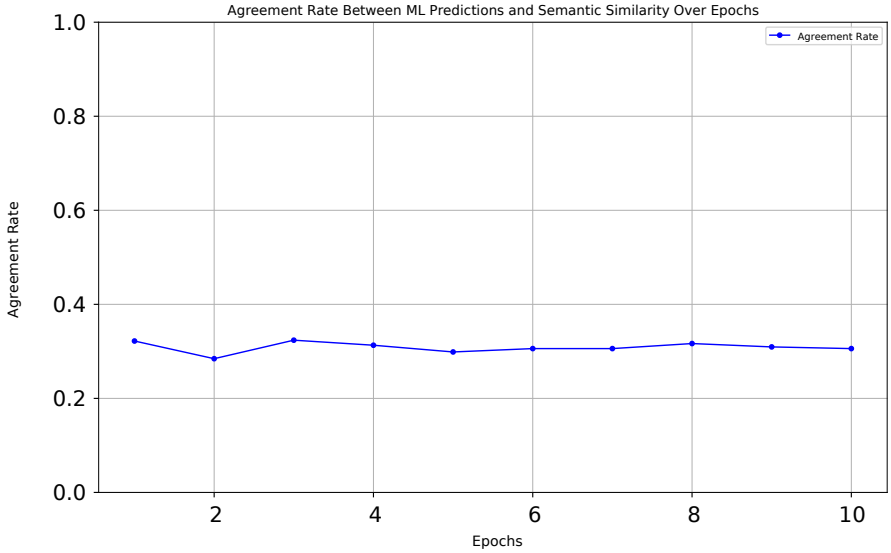


Figure 6.6: Agreement rate with Model 3

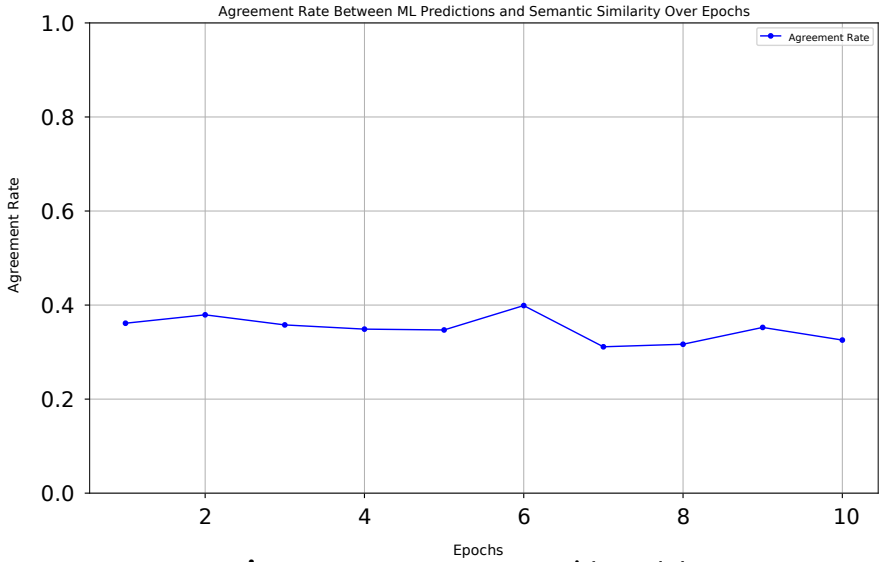


Figure 6.7: Agreement rate with Model 4

This first experiment allows us to conclude that, despite the relatively small size of the dataset (ATT&CK), the use of fine-tuning does not produce appropriate results for mappings that must be as precise as feasible.

6.4.2 Preliminary Test #2

In the second preliminary test, we applied zero-shot learning with the Facebook/bart-large-mnli and roberta-large-mnli models. We experimented with three different configurations: the first considered the *ensemble* (ZS-E) of models by averaging the class probabilities across all models; the second was a *hybrid* approach (ZS-H) by averaging semantic similarity scores and the zero-shot classification probabilities, giving equal weight to both components; and the third was a *template*, ZS-T) of the input to the model, that is, describing the context about the prediction to be implemented (it is not fine-tuning nor training).

Despite the theoretical promise of zero-shot learning, the accuracy rate is still rather low when various settings are taken into account. Both testing the whole dataset (Table 6.2) and altering the dataset size in random samples (tables 6.3, 6.4, 6.5) confirm this behaviour. These results demonstrate that zero-shot learning models are insufficient in the ATT&CK domain, even though they are explicitly designed to make inferences in situations where the input domain is unknown.

Table 6.2: ZS Learning on ATT&CK

Method	Accuracy
Zero-shot (template)	39.55
Zero-shot (hybrid)	45.31
Zero-shot (ensemble)	37.91

Table 6.3: Accuracy (%) with seed 1

-	15	20	30	45	70	100	150
ZS-T	53.33	50.0	43.33	46.67	44.29	41.0	41.33
ZS-H	53.33	55.0	50.0	48.89	48.57	45.0	45.33
ZS-E	33.33	40.0	36.67	37.78	38.57	36.0	36.67

Table 6.4: Accuracy (%) with seed 15

-	15	20	30	45	70	100	150
ZS-T	26.67	25.0	23.33	31.11	32.86	38.0	38.0
ZS-H	20.0	20.0	26.67	35.56	35.71	40.0	40.0
ZS-E	13.33	20.0	26.67	35.56	35.71	40.0	37.33

Table 6.5: Accuracy (%) with seed 35

-	15	20	30	45	70	100	150
ZS-T	13.33	30.0	26.67	35.56	38.57	37.0	36.67
ZS-H	33.33	45.0	36.67	40.0	42.86	46.0	44.0
ZS-E	26.67	40.0	33.33	35.56	37.14	37.0	38.0

This second experiment, similarly to the first one, allows us to conclude that using zero-shot learning does not produce appropriate results for mappings that need to be as precise as possible.

6.5 Mapping CAPEC To ATT&CK

The first mapping focuses on correlating CAPEC attack patterns with MITRE ATT&CK techniques. The identification and evaluation of the best correlational method is conducted by an expert, with the support of the BRON ontology. The evaluation is performed by assessing the correctness of each correlation and checking whether it is supported or confirmed by the BRON ontology entries.

We selected a random set of 15 ATT&CK techniques and let them remain fixed across the entire analysis, and four transformer-based semantic similarity models were used to identify the most relevant CAPEC attack pattern for each technique of the 15. The fixed set of techniques serves to make a consistent and fair evaluation between the models employed. Although in the general methodology we stated that we wanted to reach the techniques from the attack patterns, here we are doing the reverse, hence retrieving the attack patterns from the techniques. The motivation is that in the BRON ontology there is a mapping in this direction, and we cannot assume that the reverse also applies, i.e. that the same correlations occur.

However, in terms of semantic similarity, the result would be identical since the semantic similarity is symmetrical; therefore, the evaluation would not be affected by having considered a different direction. The same assumption is applied in the second mapping, which explores the correlations between ATT&CK techniques and the steps of the killchain. The choice to perform the evaluation on a limited set is justified by the inherently manual nature of the process and the lack of a definitive ground truth. Thanks to the limited set, we obtain methodological consistency and ease in evaluating the models' performances.

The results obtained from the partial evaluation of this first mapping are illustrated in tables 6.10, 6.7, 6.8, and 6.9. The symbols used in the tables can be found in Table 6.6, with their rationale.

Table 6.6: Legend for comparison between semantic similarity and the BRON ontology

Symb.	Rationale
✓✓	Different but correct correlations found both by SS and BRON
✓	Correct correlations found by SS but no correlation on BRON
✓	Same correct correlations found both with SS and BRON
✓	Partially correct correlation found by SS
✗	Incorrect correlation found by SS

Table 6.7: Comparison between BRON ontology and semantic similarity with model all-MiniLM-L6-v2

AP	ATT&CK ID	SS	Check
568	T1003	0.62	✓✓
220	T1071.001	0.67	✗
481	T1090.004	0.80	✓
641	T1129	0.61	✓
308	T1205.001	0.64	✓
640	T1218.010	0.56	✓
624	T1480.001	0.56	✗
131	T1496	0.65	✓
510	T1526	0.55	✓
564	T1546.013	0.57	✓
270	T1546.015	0.58	✗
645	T1558.005	0.57	✓
159	T1574	0.74	✓
309	T1595	0.70	✓✓
571	T1654	0.53	✗

Table 6.8: Comparison between BRON ontology and semantic similarity with model all-mpnet-base-v2

AP	ATT&CK ID	SS	Check
653	T1003	0.66	✓✓
33	T1071.001	0.68	✗
481	T1090.004	0.91	✓
641	T1129	0.71	✓
300	T1205.001	0.75	✓
641	T1218.010	0.65	✓
442	T1480.001	0.59	✗
125	T1496	0.66	✓
546	T1526	0.59	✗
564	T1546.013	0.48	✓
448	T1546.015	0.62	✗
652	T1558.005	0.67	✓
642	T1574	0.76	✓
309	T1595	0.74	✓✓
571	T1654	0.63	✗

Table 6.9: Comparison between BRON ontology and semantic similarity with model paraphrase-multilingual-mpnet-base-v2

AP	ATT&CK ID	SS	Check
600	T1003	0.76	✓✓
33	T1071.001	0.74	✗
481	T1090.004	0.86	✓
562	T1129	0.69	✓
300	T1205.001	0.75	✓
695	T1218.010	0.66	✗
578	T1480.001	0.66	✗
124	T1496	0.78	✗
510	T1526	0.65	✓
9	T1546.013	0.60	✓
505	T1546.015	0.68	✗
645	T1558.005	0.75	✓
30	T1574	0.82	✓
309	T1595	0.80	✓✓
647	T1654	0.75	✓

Table 6.10: Comparison between BRON ontology and semantic similarity with model attack-BERT

AP	ATT&CK ID	SS	Check
653	T1003	0.81	✓✓
662	T1071.001	0.70	✓
481	T1090.004	0.83	✓
641	T1129	0.82	✓
300	T1205.001	0.83	✓
641	T1218.010	0.61	✓
463	T1480.001	0.66	✓
130	T1496	0.81	✓
574	T1526	0.73	✓
564	T1546.013	0.64	✓
579	T1546.015	0.52	✗
645	T1558.005	0.76	✓
558	T1574	0.76	✓
309	T1595	0.70	✓✓
571	T1654	0.65	✗

As the example shows, the correlations found by the semantic similarity algorithm are mostly correct with Model 4 (attack-BERT model). In this case, only two correlations out of 15 were found incorrect. Two correlations were also doubly validated by the BRON ontology, while the others are mostly correct, with 3 having partial correctness. The most reasonable explanation is that the model is essentially a pre-trained model on the ATT&CK framework. Therefore, it is quite natural to obtain correlations that are essentially correct because the model has deep knowledge of the context of the inputs provided, i.e. attack patterns and techniques.

In the other three models, however, we can see a balance between correct and incorrect correlations. Although we can see that the semantic

similarity values are still high (therefore, rather similar terms), we can deduce that these last models probably do not capture the context as effectively as the first model. Although it may seem obvious to think of the model 4 as the best model, this is not necessarily the case. Model 4 produces good overall performance and reliable outcomes, as the above example demonstrates. In single correlations, however, it might not perform as well as other models. Technique T1654 perfectly captures this hypothesis: model 4 (trained on ATT&CK) found attack pattern 571 to be the closest match, which we marked as erroneous. Model 3, which was not trained on ATT&CK, matched the technique T1654 with attack pattern 647, which we assessed to be partially accurate. This implies that although the model 4 does better generally on big datasets, it can fall short in specific cases.

From this analysis, we can also deduce that the BRON ontology cannot be regarded as ground truth because it lacks correlations, which the application of the methods above has partially found. Considering these factors, we regard the correlations between attack patterns and techniques identified by model 4 as valid for our purposes.

6.6 Mapping ATT&CK To Any KillChain

In this mapping, by evaluating the different methods on ATT&CK as it is considered as ground truth, we establish the most accurate method in identifying tactics from techniques. Next, to derive the steps of any killchain from the techniques, we will apply the method found. Therefore, considering the first mapping and results from this second mapping, we would have a unique flow from which it is possible to reach from CAPEC's attack patterns the steps of any killchain.

The complete analysis will be divided into the next two subsections, respectively, to test the best correlational method and evaluate the obtained results (killchain steps). Therefore, the structure of this section is different from the previous one because this mapping requires a more in-depth analysis.

6.6.1 Testing

The further peculiarity compared to the previous mapping is that, in the test phase, we already know a priori which are the steps of the killchain, i.e. the ATT&CK tactics. We can assume that the steps are fixed classes into which we map the various techniques. For this reason, in addition

to the simple semantic similarity, we tested the K-Means method, which originated from clustering theory. The K-Means method still uses semantic similarity, but instead of finding *1-to-1* techniques/tactics correlations, it finds *many-to-one* correlations. The *many* techniques considered most similar by the semantic similarity models are correlated with the *1* most similar tactic.

For reasons of visualisation of the following tables, we have associated acronyms with the names of the methods used (Table 6.11). In the graphs, the respective methods will take on the colours given in the Figure 6.8.

Table 6.11: Legend of semantic similarity and clustering acronyms

Method	Name
SS (Model 1)	A1
KMeans (Model 1)	A2
SS (Model 2)	B1
KMeans (Model 2)	B2
SS (Model 3)	C1
KMeans (Model 3)	C2
SS (Model 4)	D1
KMeans (Model 4)	D2

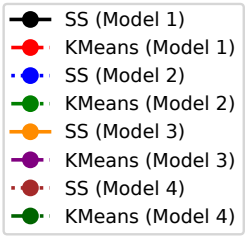


Figure 6.8: Legend of colours

Since the evaluation is performed on a ground truth (ATT&CK), we evaluate the methods using a typical machine learning metric, the *accuracy*. We only consider accuracy, in percentage terms, because we are only interested in how accurate the answers provided in the output by the various methods are, i.e. how correct they are considering the ground truth. We regard not strictly necessary, considering other metrics, since our objective is to find the best correlational method, i.e. the method that most correctly predicts a killchain step, without particular attention to false positives and false negatives.

We considered two macro categories of experimental settings. In the first category, we applied the semantic similarity to the whole ATT&CK. In the first instance, we applied the semantic similarity to ATT&CK as it is. In the second instance, we have applied the semantic similarity by considering a *combinatorial setting* of ATT&CK. In particular, we can observe that each ATT&CK technique description is distributed into several paragraphs, the quantity of which is rather variable and not fixed a priori. However, some paragraphs may be semantically closer to a tactic description, others less. The description of a tactic is quite abstract, while some descriptions of techniques may contain descriptions of specific settings and components of systems to which the specific technique applies, or to be able to replicated. Consequently, these paragraphs may negatively influence the semantic correlation because we try to correlate a specific description with an abstract one, causing accuracy to drop

accordingly.

In this first category, the application of the various models, only with semantic similarity, is shown in Table 6.12 and Table 6.13.

Table 6.12: SS accuracy on full ATT&CK

Method	Accuracy (%)
SS (Model 1)	58.33
SS (Model 2)	57.39
SS (Model 3)	39.08
SS (Model 4)	72.18

Table 6.13: SS accuracy on full ATT&CK (best paragraph combinations)

Method	(Accuracy)
SS (Model 1)	59.86
SS (Model 2)	56.92
SS (Model 3)	39.08
SS (Model 4)	68.90

From the results, we can immediately notice that the application of semantic similarity alone tends to yield better results than fine-tuning a model and zero-shot learning, as seen in preliminary tests in which both are involved. The highest accuracy percentage is obtained by applying model 4, trained on ATT&CK, where we can say that we have obtained quite satisfactory accuracy percentages. The other models, however, show a significant increase, albeit minimal, compared to the use of fine-tuning, where the accuracy percentage in terms of inference was around 40%. The effect of choosing the combinatorial setting is visible only in model 1, where there is a slight increase of around 1.5% For the other models, paradoxically, we can see how the combinatorial setting yields a slightly lower overall accuracy compared to the normal case, where the entirety of a paragraph of a technique is given as input.

Next, as the second category of experimental setting, we focus our attention on the accuracy gained by applying each method to a random sample of techniques. Instead of evaluating the accuracy of the methods on the entirety of ATT&CK, we evaluate the application of methods on random samples, gradually incremental with size ranging from 15 to 100, and with increments between 5 and 50 units. We replicated the same experiment by changing the seed for the randomness of the sample to verify how much the accuracy varies, and to verify that, if a model is more performant than others, then it is consistently so when the samples vary.

We provide table-graph pairs, where the results are shown first and then plotted after. The pairs are: Table 6.14 and Figure 6.9, Table 6.15 and Figure 6.10, Table 6.16 and Figure 6.11.

Table 6.14: SS Accuracy (%) - first seed

-	15	20	30	45	70	100	150
A1	80.0	80.0	70.0	55.56	57.14	55.0	57.33
A2	80.0	85.0	56.67	40.0	54.29	34.0	35.33
B1	60.0	60.0	60.0	60.0	57.14	57.0	60.0
B2	60.0	70.0	53.33	48.89	30.0	40.0	27.33
C1	46.67	40.0	40.0	31.11	32.86	33.0	36.0
C2	46.67	25.0	23.33	17.78	14.29	17.0	24.67
D1	80	75.0	73.33	68.89	67.14	73.0	70.0
D2	80	70	56.67	48.89	51.43	54.0	51.33

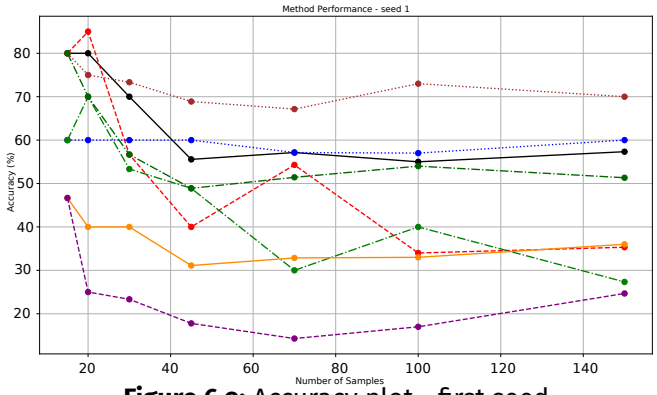


Figure 6.9: Accuracy plot - first seed

Table 6.15: SS Accuracy (%) - second seed

-	15	20	30	45	70	100	150
A1	73.33	65.0	63.33	64.44	61.43	65.0	63.33
A2	66.67	60.0	66.67	51.11	44.29	42.0	52.0
B1	53.33	55.0	56.67	60.0	58.57	60.0	58.67
B2	53.33	65.0	50.0	60.0	37.14	44.0	50.0
C1	46.67	40.0	30.0	37.78	32.86	36.0	36.0
C2	40.0	40.0	26.67	28.89	18.57	27.0	22.0
D1	86.67	75.0	73.33	77.78	74.29	75.0	74.0
D2	86.67	70.0	63.33	71.11	62.86	67.0	62.0

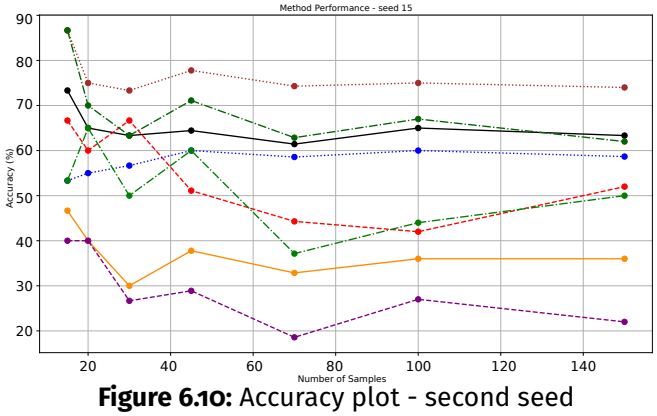


Figure 6.10: Accuracy plot - second seed

Table 6.16: SS Accuracy (%) - third seed

-	15	20	30	45	70	100	150
A1	66.67	65.0	60.0	60.0	67.14	70.0	64.67
A2	66.67	55.0	50.0	46.67	55.71	47.0	51.33
B1	46.67	50.0	50.0	57.78	57.14	60.0	58.0
B2	46.67	50.0	56.67	48.89	47.14	45.0	46.67
C1	26.67	30.0	33.33	33.33	37.14	42.0	40.67
C2	26.67	25.0	20.0	15.56	20.0	20.0	24.0
D1	80.0	80.0	83.33	82.22	81.43	80	78.67
D2	80.0	85.0	73.33	73.33	67.14	63.0	56.0

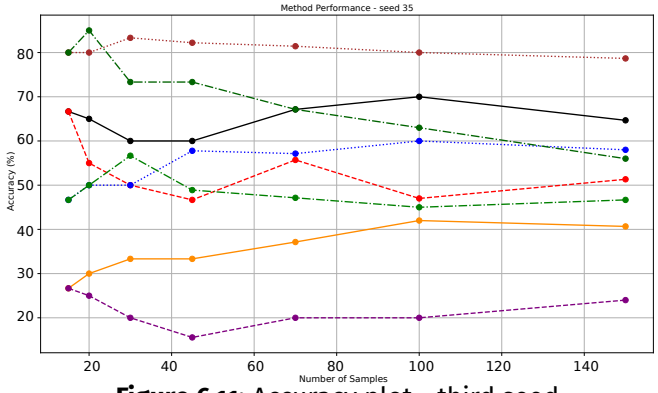


Figure 6.11: Accuracy plot - third seed

This analysis aims to identify the best correlational method by evaluating its performance relative to variations in dataset size, specifically, changes in the number of input samples. We can say that, as a general analysis, if smaller samples are considered than the full input set, i.e. the entire ATT&CK, certain methods perform better, while others show only marginally increased accuracy.

Among the tested methods, Model 4 consistently achieves the highest accuracy, approaching 90%. Across the three different random seeds, its accuracy does not fall below 70%, when evaluated on the largest sample size too. Also, model 1 yields results considered promising by reaching an accuracy of 80% with the first seed and first two sample sizes. However, this performance is inconsistent across other seeds. The high accuracy may be attributed to a particularly favourable random sample. This observation is reinforced as the sample size increases. However, although accuracy declines, it remains close to 60% even in the worst-case sample size. These findings suggest that models not specifically trained on security-focused data can still perform well in correlational tasks. Similarly, the application of K-Means clustering to both models exhibits a comparable trend, with accuracy levels remaining generally satisfactory. Models 2 and 3, by contrast and despite being trained on a broader range

of information, fail to effectively capture the security domain context. Their performance is noticeably lower, as a consequence. Model 2 reaches a peak accuracy of 60% when only applied to the smallest sample size (15), while Model 3 performs the worst overall, with accuracy dropping to a minimum of 22% with K-Means methods.

In case one has a sufficiently large set of techniques, our analyses suggest that instead of applying the best correlational method (in this case semantic similarity with model 4) on the entire set, it is reasonable to apply it on small samples, even random ones, to increase the correctness on the predictions of the tactics, and consequently on steps of any killchain.

6.6.2 Evaluation

Once the best correlational method has been found, this section deals with an evaluation of the results that can be obtained. That is, applying the method no longer to predict ATT&CK tactics, but the steps of the other considered killchains. The analysis of this method is fundamental because we have noticed how conceptually similar the steps of different killchains are, even if the killchains may have a different number of steps. Therefore, we assume that applying the best correlational method to find the steps provided by ATT&CK would be *logically equivalent* to applying the best correlational method to predict the steps of any killchain. This simple assumption is translated into the following procedure, illustrated in Algorithm 7.

Algorithm 7: Technique-to-Tactic Correlation Algorithm

Input: ATT&CK technique T, killchain X
Output: Step R

```
if T ∈ X then
    R ← tactic associated with T;
else
    Apply SS method;
    R ← result of SS method;
end
return R;
```

The algorithm has the following rationale. Given an ATT&CK technique T and a killchain X, if X contains the same step R (or tactic, considering the ATT&CK side) related to technique T, then return the

shared step *R*. Otherwise, apply the best correlational method found for returning the most correlated step.

By applying this procedure, leveraging the already used techniques' sample of the first mapping, we obtain the correlations illustrated in Table 6.17 and Table 6.18 for, respectively, retrieving steps from Lockheed-Martin killchain and Unified killchain.

Table 6.17: ATT&CK techniques associated with Lockheed-Martin killchain steps

ATT&CK ID	Obtained LM Step	Original Tactic	Tactic Sim.
T1003	Installation	Persistence	0.59
T1071.001	Command & Control	Command and Control	1
T1090.004	Delivery	Execution	0.56
T1129	Installation	Persistence	0.59
T1205.001	Command & Control	Command and Control	1
T1218.010	Installation	Persistence	0.59
T1480.001	Weaponization	Execution	0.49
T1496	Exploitation	Initial Access	0.50
T1526	Reconnaissance	Reconnaissance	1
T1546.013	Installation	Persistence	0.59
T1546.015	Installation	Persistence	0.59
T1558.005	Installation	Persistence	0.59
T1574	Installation	Persistence	0.59
T1595	Reconnaissance	Reconnaissance	1
T1654	Reconnaissance	Reconnaissance	1

Table 6.18: ATT&CK techniques associated with Unified killchain steps

ATT&CK ID	Obtained Unified Step	Original Tactic	Tactic Sim.
T1003	Credential Access	Credential Access	1
T1071.001	Pivoting	Command and Control	0.55
T1090.004	Pivoting	Command and Control	0.55
T1129	Execution	Execution	1
T1205.001	Pivoting	Command and Control	0.55
T1218.010	Execution	Execution	1
T1480.001	Defense Evasion	Defense Evasion	1
T1496	Resource Development	Resource Development	1
T1526	Reconnaissance	Reconnaissance	1
T1546.013	Persistence	Persistence	1
T1546.015	Persistence	Persistence	1
T1558.005	Credential Access	Credential Access	1
T1574	Execution	Execution	1
T1595	Discovery	Discovery	1
T1654	Collection	Exfiltration	0.69

As can be seen from the tables, the best correlational method, for each killchain different from ATT&CK found different steps from the selected killchains. Since there is no ground truth to accurately validate the obtained correlations, and since the number of correlations themselves is quite manageable, we perform what we call a *partial validation*, or a manual validation.

Lockheed-Martin killchain In the case of the Lockheed-Martin killchain steps (Table 6.17), we can see how the correlations, in the case the steps are not identical, are quite similar. There are mainly three:

- **Installation correlated with Persistence**. The two steps are identical; we can obtain *persistence* on a system if we *install* malicious software.
- **Exploitation correlated with Initial Access**. The two steps are identical: we can gain *initial access* to a system if we *exploit* known vulnerabilities.
- **Weaponization correlated with Execution**. The two steps are conceptually close: we can *execute* attacks if we *weaponize* some initially harmless content, making it malicious.

Unified killchain Even in the case of the Unified killchain steps (Table 6.18), we can see how the correlations are quite similar. There are mainly two:

- **Pivoting correlated with Command and Control**. Two steps are conceptually close: we can *pivot* a network unit to *command and control* the rest of the network.
- **Collection correlated with Exfiltration**. Two steps are identical: we can obtain *information collection* through *exfiltration* of improperly appropriated content.

6.7 Mapping Security Measures To Killchains

In this last mapping, we illustrate how to combine the previous two mappings to obtain a complete flow that, starting from security measures, can create structured and ad-hoc killchains. So far, however, we have not mentioned any security measures, although the working hypothesis is that killchains can be built from the security measures themselves.

By backwards applying `WISARD`, we can arrive at some security measures and hypothesise that the latter are measures to which an organisation has resulted non-compliant. We identify the security measures illustrated in Table 6.19, which are respectively correlated with the attack patterns shown in Table 6.20. The measures are derived from the NIS 2 Directive, specifically from Articles 7 and 21.

Table 6.19: Excerpt of NIS 2 measures

Measure ID	NIS 2 Measures
A	*strengthening the cyber resilience and the cyber hygiene baseline of small and medium-sized enterprises, in particular those excluded from the scope of this Directive, by providing easily accessible guidance and assistance for their specific needs*
B	*policies and procedures regarding the use of cryptography and, where appropriate, encryption*
C	*managing vulnerabilities, encompassing the promotion and facilitation of coordinated vulnerability disclosure under Article 12(1)*
D	*business continuity, such as backup management and disaster recovery, and crisis management*
E	*incident handling*
F	*the use of multi-factor authentication or continuous authentication solutions, secured voice, video and text communications and secured emergency communication systems within the entity, where appropriate*
G	*related to sustaining the general availability, integrity and confidentiality of the public core of the open internet, including, where relevant, the cybersecurity of undersea communications cables*

Table 6.20: Measures correlated with respective attack patterns

M. ID	*A*	*G*	*G*	*B*	*G*	*C*	*F*	*E*	*D*	*F*	*E*	*F*	*G*
AP ID	130	300	309	463	481	558	564	571	574	579	641	645	662

By taking into account the security measures, in the flow presented in Figure 6.2, attack patterns act as a connection point between security measures (left side) and killchain steps (right side). The right side is entirely accessible via the mappings discussed in this chapter. The entire flow we want to build is based on this consideration: assuming that the organisation was not compliant with a set of measures, by applying the mappings seen previously, we have a sequential and ordered flow of attack steps, where each step is composed of attack patterns. In other words, we can say that using *killchain step X*, exploiting *attack pattern Y*, we can attack the *security measure Z*. Thus, in terms of auditing or penetration testing, if some tests were being carried out, and there was also knowledge of the compliance situation, the latter could be exploited to the attacker's advantage.

Considering the set of security measures introduced in this last mapping, by considering the previous mappings, for the Lockheed-Martin and Unified killchains, we obtain the attack flows respectively illustrated in Table 6.21 and Table 6.22.

Table 6.21: Lockheed-Martin killchain steps associated with the NIS 2 security measures

Employing Step...	#Step	Leveraging AP...	For Attacking Measure...
Reconnaissance	1	574	D
Reconnaissance	1	571	E
Reconnaissance	1	309	G
Weaponization	2	463	B
Delivery	3	481	G
Exploitation	4	130	A
Installation	5	579	F
Installation	5	645	F
Installation	5	564	F
Installation	5	641	E
Installation	5	653	F
Installation	5	558	C
Command and Control	6	662	G
Command and Control	6	300	G

Table 6.22: Unified killchain steps associated with the NIS 2 security measures

Employing Step...	#Step	Leveraging AP...	For Attacking Measure...
Reconnaissance	1	574	D
Resource Development	2	130	A
Persistence	6	579	F
Persistence	6	564	F
Defence Evasion	7	463	B
Pivoting	9	481	G
Pivoting	9	300	G
Pivoting	9	662	G
Discovery	10	309	G

Execution	12	641	E
Execution	12	641	E
Execution	12	558	C
Credential Access	13	645	F
Credential Access	13	653	F
Collection	15	571	E

Starting from the initial set of 15 techniques identified in the first mapping phase, and leveraging them backwards to refine and stabilise MOSKAD through the various mappings, it becomes possible to construct nearly complete and structurally coherent killchains.

Upon analysing the resulting killchains, a higher degree of completeness is achieved by the Lockheed-Martin killchain. Specifically, steps 1 through 6 are fully represented, often with multiple attack patterns corresponding to the same step. This suggests that different attack patterns can be used for the same step. Contextually, the final two steps of the Lockheed-Martin killchain remain unaddressed, likely due to the inherent limitations of the sample size, i.e. lack of techniques that could be associated with the two uncovered tactics.

Instead, when considering the Unified killchain — we recall it consists of 18 distinct steps —, a natural limitation arises due to the larger number of steps in comparison with the 15 techniques of the sample. Therefore, the maximum number of unique steps that can be potentially covered is likewise limited at 15, leaving at least 3 steps inevitably unrepresented, regardless of the diversity of the selected techniques. This limitation is not structural but rather dependent on the chosen sample size. If the goal is to achieve a broader or even complete coverage of the Unified killchain, this could be achieved simply by increasing the number of input techniques. For example, extracting a sample of 20 techniques — only if from the security measures it is possible to reach a sufficient variety of attack patterns to support this assumption — would increase the probability of mapping to a larger number of killchain steps, thus improving coverage and representativeness. The role of mapping 2 is that, theoretically, all steps can be traversed dynamically, corresponding to the attack patterns derived from the security measures.

6.8 Automating MOSKAD

The automation of MOSKAD relies on the automation of the different mappings. Starting from the last one, it is conducted by WISARD – then the same considerations already discussed go. Instead, the remaining mappings both combine automatic with manual components.

We consider automatic components the employment of the different tools for obtaining different correlations. Machine learning for the preliminary tests, semantic similarity, and clustering automatically derive the requested associations.

The evaluation of the mappings is also to be considered automatic, thanks to the availability of the BRON ontology (for evaluating the first mapping) and to the structure of the ATT&CK framework, which makes it a ground truth (for evaluating the second mapping). However, when dealing with a reasonable number of correlations, a manual validation has been performed to evaluate the correctness of the tools employed.

6.9 Concluding Remarks

Non-compliance with the security measures prescribed in security directives may leave companies exposed to weaknesses, which consequently can lead to attacks. An attacker, or penetration tester, may leverage this new source of information for targeted security activities.

In this chapter, we propose how to build structured attack flows — namely, killchains — tailored to the security measures a company has missed. For achieving this, we designed `MOSKAD`, a methodology with three different mappings, beginning with the attack patterns defined in CAPEC and leveraging the ATT&CK framework as an intermediate layer for retrieving the steps of any killchain.

We tested different machine learning techniques to find the *best correlational method*, i.e. the most performing method for executing each mapping. Preliminary tests highlighted how fine-tuning and zero-shot learning yield insufficient accuracy for the considered mappings, leaving the semantic similarity to be the most effective candidate.

Upon considering a set of techniques from ATT&CK through the whole methodology, we showed how correlating the techniques with CAPEC's attack patterns, and how correlating the techniques with the steps of Lockheed-Martin and Unified killchain, as possible examples of killchains.

Chapter 7

Setàd: Leveraging WISARD Semantic Similarity Step for Correlating Non-Security Directives Sources

What we call chaos is just patterns we haven't recognised. What we call random is just patterns we can't decipher.

— Chuck Palahniuk

This chapter introduces Setàd as a methodology for re-applying the algorithms from the semantic similarity step of WISARD, aimed at deriving attack tactics from attack techniques. The study focuses on the MITRE ATT&CK framework, which maps concrete attack techniques to high-level adversarial tactics, providing a comprehensive benchmark of the main machine learning metrics. It assesses the effectiveness of Setàd in capturing the semantic relationships between techniques and their associated tactics.

7.1 Introduction

Computer security has become such a common phenomenon that even the attack methods employed by the most famous threat groups have become predictable. Even among different groups, these methods have become so similar that they have led the scientific community to classify, taxonomise and publicly document them. Several taxonomies have arisen, both for attack [93] and defence [94], and lastly, concerning artificial intelligence attacks [91]. These taxonomies aim to provide a specific and detailed perspective of an attacker's methods against specific systems, networks, or protocols.

Among the most widely adopted frameworks is MITRE's ATT&CK [92], which systematically categorises adversarial methods. ATT&CK structures these methods into *techniques* and *tactics*. A technique describes a specific action used by an attacker, e.g. phishing, rootkit, etc. A tactic describes the specific goal an attacker aims to accomplish, such as initial

access to systems or privilege escalation. Each technique corresponds to a specific tactic — a tactic is also a *step* of the killchain ATT&CK itself defines — and a tactic corresponds to multiple techniques.

In this chapter, we discuss and benchmark `Setàd` (Semantic tactics derivation), a methodology for semantically deriving attack tactics from attack techniques. `Setàd` employs the families of algorithms designed in the semantic similarity step of `WISARD`. An evaluation of the algorithms is performed in terms of precision, recall, F1-score and accuracy, using the ATT&CK framework itself as ground truth.

7.2 Related Work

Our state-of-the-art research has not led to any work that is similar to ours, i.e. that considers ATT&CK as ground truth and evaluates possible mapping tools. Therefore, our related work section is limited to papers where the importance of ATT&CK in the industrial and academic context is emphasised.

Arafune et al. [12] present an automated threat hunting framework based on ATT&CK. The authors particularly focus on threat hunting in ICS (Industrial Control Systems) networks. Rodriguez et al. [115] propose a method by combining process mining with ATT&CK. The method aims to categorise observed attack strategies to process models. The authors demonstrated the method on particularly crafted human-behaviour-related attacks. Zambianco et al. [144] leverage ATT&CK framework as an optimisation-based decoy selection scheme for accounting attack preconditions and effects. Pell et al. [108] address adversarial TTPs on 5G networks by extending ATT&CK to include technologies peculiar to 5G, such as SDN and signalling protocols. Rosso et al. [117] introduce SAIBERSOC, a methodology for performance evaluation of security operation centres. The authors use synthetic attacks mapped to ATT&CK. Kuppa et al. [75] Employ cosine similarity as a component of a more intricate design to automatically map CVEs to ATT&CK methods. CVEs are semantically enhanced to make it easier to connect them to known attacks and defences because of their high degree of depth and specificity. Noor et al. [98] utilise the semantics idea for many phases in a flow that begins with the identification of tactics, methods, and procedures (TTPs) to map and classify them into Cyber Threat Intelligence Reports (CTIRs) and, ultimately, into (Threat TTP-Detection) TTDs using ATT&CK techniques. They employ latent semantic indexing (LSI) instead of cosine similarity.

Similar investigations have been conducted in the following two works. Roy et al. [118] present a taxonomic systematisation of the research literature on ATT&CK by studying its degree of usefulness in different applications. Al-Sada et al. [10] inspected more than 50 primary research works about ATT&CK. The authors provided a categorisation of different adopted methodologies.

7.3 The Setàd Methodology

Setàd reuses the two families of algorithms previously introduced within the semantic similarity step in WISARD. However, their inclusion here is justified by their formalisation in Isabelle, which provides a rigorous and machine-checked representation. This formal treatment not only strengthens the correctness arguments made earlier but also enables further reasoning and verification within the Isabelle/HOL framework. Furthermore, formalisation rethinks the syntax of algorithms in light of the new context analysed here.

```
text < Algorithm 1: AMax (Absolute Max)
  Algorithm 2: RMax (Relative Max) >
```

To formalise the input, we associate techniques, tactics, and models with sets of natural numbers, respectively called *Te*, *Ta*, and *SSModels*.

```
consts
  Te :: "nat set" (* Techniques *)
  Ta :: "nat set" (* Tactis *)
  SSModels :: "nat set" (* Semantic similarity models *)
```

We also define the *SS* function for calculating the semantic similarity, which takes three inputs, a technique, a tactic, and a model (as natural numbers), and returns the cosine similarity value (real value).

```
  consts SS :: "nat ⇒ nat ⇒ nat ⇒ real"
```

At the same time, we also define the set *SS_results*, which includes all semantic similarity values calculated with all techniques, tactics and models.

```
definition SS_results :: "real set" where
  "SS_results = {SS ta te s | ta te s. ta ∈ Ta ∧ te ∈ Te ∧ s ∈
     SSModels}"
```

To retrieve the semantic similarity values calculated within a single model, we define the *SS_model* set. It takes as input a model (natural

number) and returns the semantic similarity values calculated within the model (set of real numbers).

```
definition SS_results_model :: "nat ⇒ real set" where
"SS_results_model s = {SS te ta s | te ta s. te ∈ Te ∧ ta ∈ Ta ∧ s ∈
    SSModels}"
```

The first family of algorithms, *AMax* (Absolute Max), has the following rationale. For each technique *te* there exists a tactic *ta* and a subset of models *s* such that, let *x* be the max of the semantic similarity values, the semantic similarity value between *te* and *ta* for each model *ss* in the subset *s* is equal to *x*. This indicates that the absolute max value converges on the same technique and tactic.

By specifying the cardinality of *s*, we consider two cases. In the case of *AMax0*, the cardinality of *s* is equal to the number of models. In the case of *AMax1*, the cardinality of *s* is variable and ranges from 1 to the number of models.

```
axiomatization where
  AMax0: "∀te ∈ Te. ∃ta ∈ Ta. ∃s ⊆ SSModels.
         (∃x. x = Max (SS_results) ∧
          (∀ss ∈ s. SS te ta ss = x) ∧ (card s = card SSModels))"
```

```
axiomatization where
  AMax1: "∀te ∈ Te. ∃ta ∈ Ta. ∃s ⊆ SSModels.
         (∃x. x = Max (SS_results) ∧
          (∀ss ∈ s. SS te ta ss = x) ∧ (1 ≤ card s ∧
          card s ≤ card SSModels))"
```

The second family of algorithms, *RMax* (Relative Max), has the following rationale. For each technique *te* there exists a tactic *ta* and a subset of models *s* such that, for each model *ss* let x_i be the max semantic similarity value calculated on *ss*, x_i occurs in the same tactic *ta*. This indicates that the relative max values of each model converge on the same technique and tactic.

By specifying the cardinality of *s*, we consider two cases. In the case of *RMax2*, the cardinality of *s* is equal to the number of models. In the case of *RMax3*, the cardinality of *s* is variable and ranges from 1 to the number of models.

```
axiomatization where
  RMax2: "∀te ∈ Te. ∃ta ∈ Ta.
             (∃s ⊆ SSModels. (∀ss ∈ s.
                 (∃xi. xi = Max (SS_results_model ss) ∧ (SS te ta ss =
                     xi))) ∧
```

```
5                   (card s = card SSModels))"
```

```
1 axiomatization where
2   RMax3: "∀te ∈ Te. ∃ta ∈ Ta.
3               (∃s ⊆ SSModels. (∀ss ∈ s.
4                   (∃xi. xi = Max (SS_results_model ss) ∧ (SS te ta ss =
                        xi))) ∧
5                   (1 ≤ card s ∧ card s ≤ card SSModels))"
```

To prove the properties of the AMax0 algorithm, we conceptualised the following lemma, which is also applicable to other cases. The lemma consists of two parts: the first part indicates the preconditions, i.e. the formalisation of the algorithm; the second part suggests its application. In the first part, we modelled the AMax0 algorithm, as we did before. In the second part, we proved the properties of AMax0 by indicating the existence of a subset *s1*. That is, *s1* is a possible subset of models in which the max conditions defined in the algorithm are satisfied.

```
1 AMax_lemma "∀ te ∈ Te. ∃ ta ∈ Ta. ∃ s ⊆ SSModels. ∃ x. x = Max
      (SS_results) ∧
2     (∀ ss ∈ s. SS te ta ss = x) ∧ (1 ≤ card s ∧ card s ≤ card
          SSModels) ⇒
3         ∃ s1 ⊆ SSModels. (∀ ss1 ∈ s1. SS te ta ss1 = x ∧ card s1 = card
              SSModels)"
4 by auto
```

The application of the algorithms involves two cases, a *static case* and a *recursive case*. The static case correlates a single technique with a single tactic. The recursive case may correlate a single technique with multiple tactics. The recursive case is handy because in ATT&CK, there are techniques associated with multiple tactics. It can be verified that the maximum amount of additional mappings is at most 2, i.e. one technique is associated with at most two tactics.

7.4 Benchmarking Setàd

In this section, we benchmark Setàd in classifying tactics associated with ATT&CK techniques, i.e. the formalised families of algorithms. We have considered 4 different models, so the analyses will be presented concerning the variability of the number of models used (later called *settings*), as previously described for each algorithm. We employed the same models as WISARD: all-Minilm-L6-v2 (model 1), all-mpnet-base-v2 (model 2), paraphrase-multilingual-mpnet-base-v2 (model 3), and attack-

BERT (model 4). In this chapter, we propose a benchmarking based on ATT&CK v16 (version 16). The techniques are processed in alphabetical order by their names (rather than IDs), covering the range from T1548 (Abuse Elevation Control Mechanism) to T1220 (XSL Script Processing).

We provide different statistics concerning precision, recall, F1-score and accuracy, whose mathematical formulation has already been presented within the WISARD methodology. Initially, the statistics are general, progressively narrowing the focus to more detailed ones. Firstly, we present histograms depicting the frequency of the metrics considering the interval [0,1]. Then, we present a cumulative benchmarking of the metrics throughout the techniques. Finally, we present a non-cumulative benchmarking of the metrics, again throughout the techniques.

We illustrate six scenarios with six different diagrams, from top left to bottom right respectively, represented: AMax1 with one model, AMax1 with two models, RMax3 with 1 model, RMax3 with 2 models, RMax3 with 3 models, RMax2 with 4 models. For each scenario, we analyse both the static and recursive cases.

7.4.1 Histograms of metrics

The first benchmarking concerns the general behaviour of the calculated metrics. We present histograms that indicate, in quantitative terms and relative to the specific value obtained, the numerosity of the individual metrics. This benchmarking allows us first to understand how much, for each algorithm, the metrics take on satisfactory values, i.e. values that are closer to one.

Static Case The first benchmarking concerns the static case, i.e. without the application of recursion. The histograms in Figure 7.1 and Figure 7.2 show how the metrics are instead distributed, and in general, we would say evenly distributed. In almost all settings, we can see a kind of balance between values that are close to zero and values that are close to one. The intermediate values, on the other hand, are smaller in quantitative terms. This indicates that the particular setting considered, in its generality, either performs very well (i.e. values close to 1) or performs very poorly (i.e. values close to zero) in identifying the correct mappings.

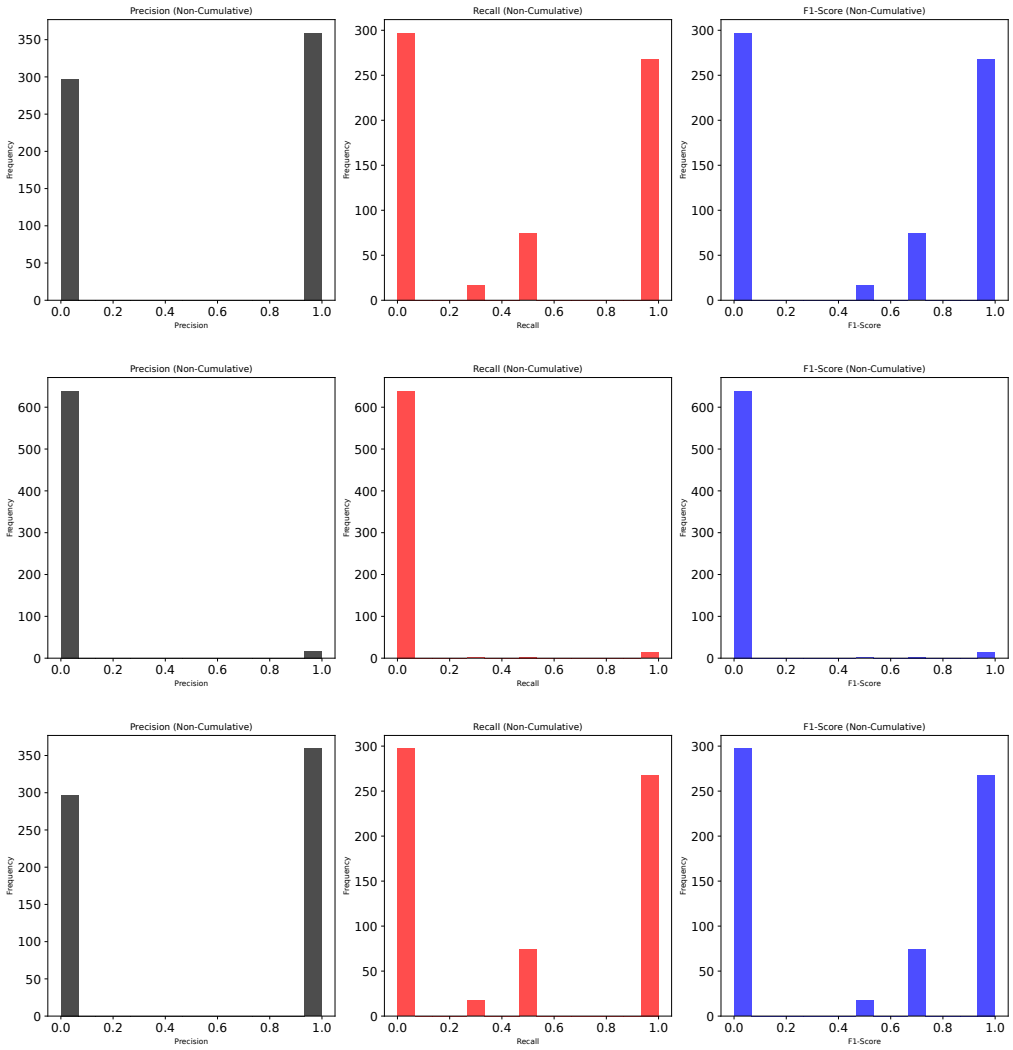


Figure 7.1: Part 1: From top to bottom: application of the semantic similarity algorithms in the static case (6 scenarios) and calculation of the number of occurrences of each machine learning metric within the [0,1] range (first 3 scenarios)

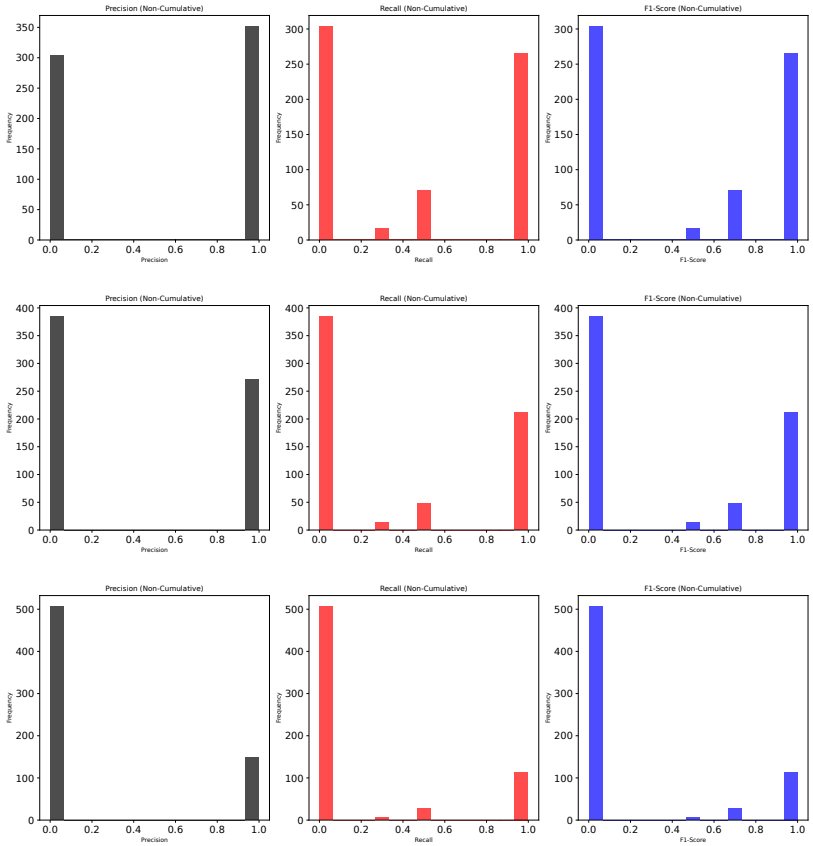
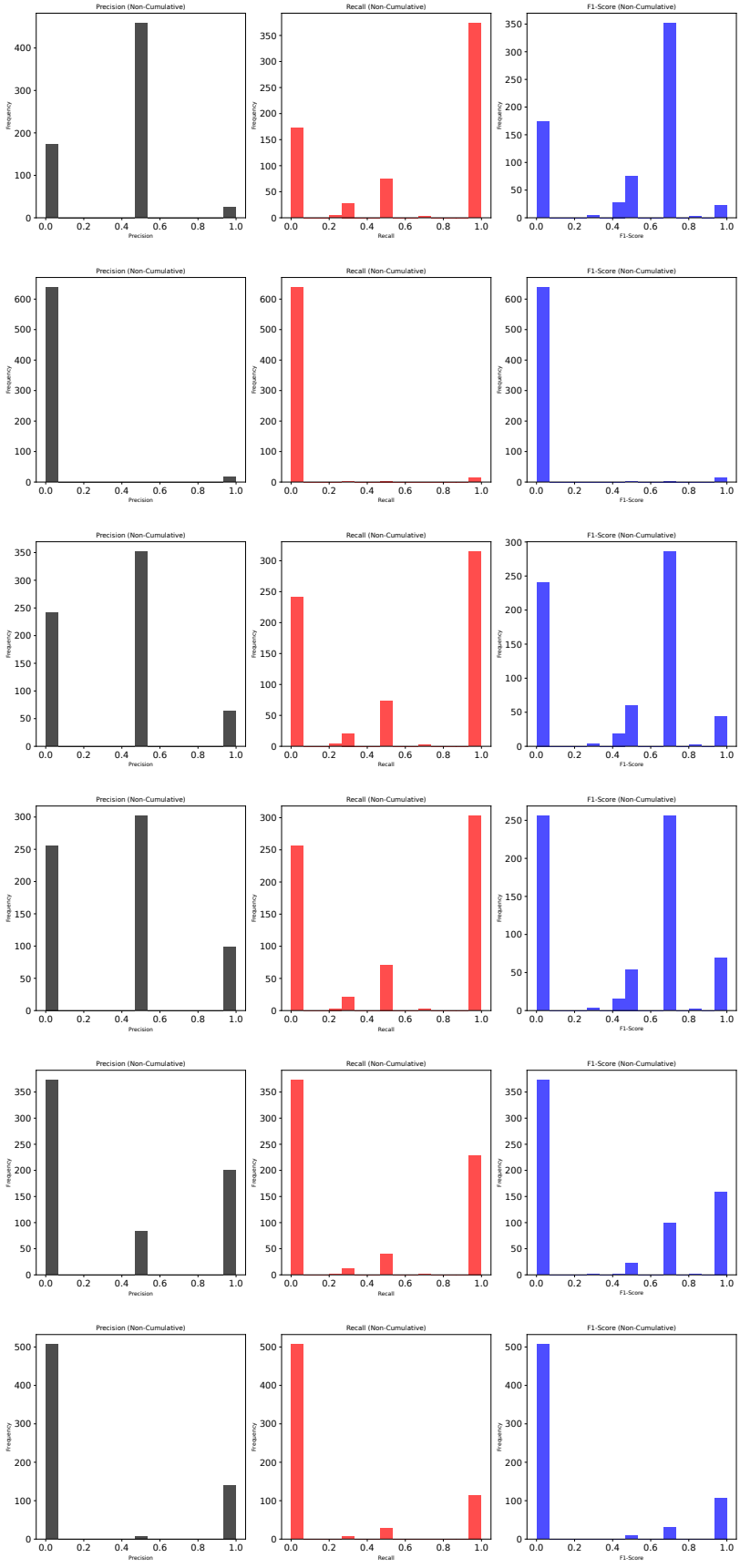


Figure 7.2: Part 2: From top to bottom: application of the semantic similarity algorithms in the static case (6 scenarios) and calculation of the number of occurrences of each machine learning metric within the [0,1] range (remaining 3 scenarios)

Recursive Case The second benchmarking concerns the recursive case, i.e. the application of the recursion. Analysing the histograms in Figure 7.3, on the other hand, we can see that the balance found in the previous case is somewhat alleviated, and the opposite situation considered occurs, i.e. the metrics are more frequent in the intermediate values. This indicates that recursion, i.e. the search for additional tactics related to the same technique, lowers the performance of semantic similarity. The cases that are negatively affected are mainly two: when a technique is effectively correlated with more than one tactic, the algorithms may obtain a wrong correlation; when a technique is not effectively correlated with more than one tactic, the algorithms may correlate the technique with non-correlated tactics.

Figure 7.3: From top to bottom: application of the semantic similarity algorithms in the recursive case (6 scenarios), and calculation of the number of occurrences of each machine learning metric within the [0,1] range



7.4.2 Cumulative plots

This section analyses and shows the behaviour of the metrics in cumulative terms. This benchmarking shows us what the trend is in the distribution of metrics when considering techniques incrementally. It then allows us to visualise the relationship that exists between the metrics, the assumed values (also helpful in considering thresholds), and how much the metrics calculated on the individual technique contribute to improving or worsening the trend of the particular metric.

Static Case Considering the static case, illustrated in Figure 7.4, we can draw different conclusions for each algorithm. In the case of the AMax1 algorithm, with 1 model, in the first half of the techniques included in the interval [0, 100], there are F1-score and recall values that differ by more than 0.35 compared to the highest precision point, and with low relative values of F1-score and recall. Starting from the second half of the same interval, the values of F1-score and recall increase until they stabilise at a value of 0.5 starting from the first part of the interval [300, 400]. The precision of the distance also fades, with a difference of at most 0.1. So, considering the maximum obtained in a single model gives mappings that are half accurate (good balance of true positives, false positives and false negatives). The AMax1 algorithm, with two models, performs significantly worse, with peaks of 0.1 for precision, and a stable, albeit fluctuating, trend between 0.02 and 0.04 for all metrics. The RMax3 algorithm, with two models, behaves almost identically to the RMax3 algorithm with 1 model (and consequently AMax1 with 1 model). The behaviour of the metrics is essentially identical, with the curves being only slightly different (touching slightly lower values) at some points on the graph. The RMax3 algorithm, starting with three models, also presents a collapse of the values in the first half of the interval [0,100]. Values that stabilise much earlier than the AMax1 algorithm, with two models, but that remain in a lower range of values, between 0.5 and 0.4. The RMax2 algorithm with 4 models presents the same characteristics, but here the range of values is further reduced, with stability of the curves obtained between 0.4 and 0.2, with a tendency towards 0.2.

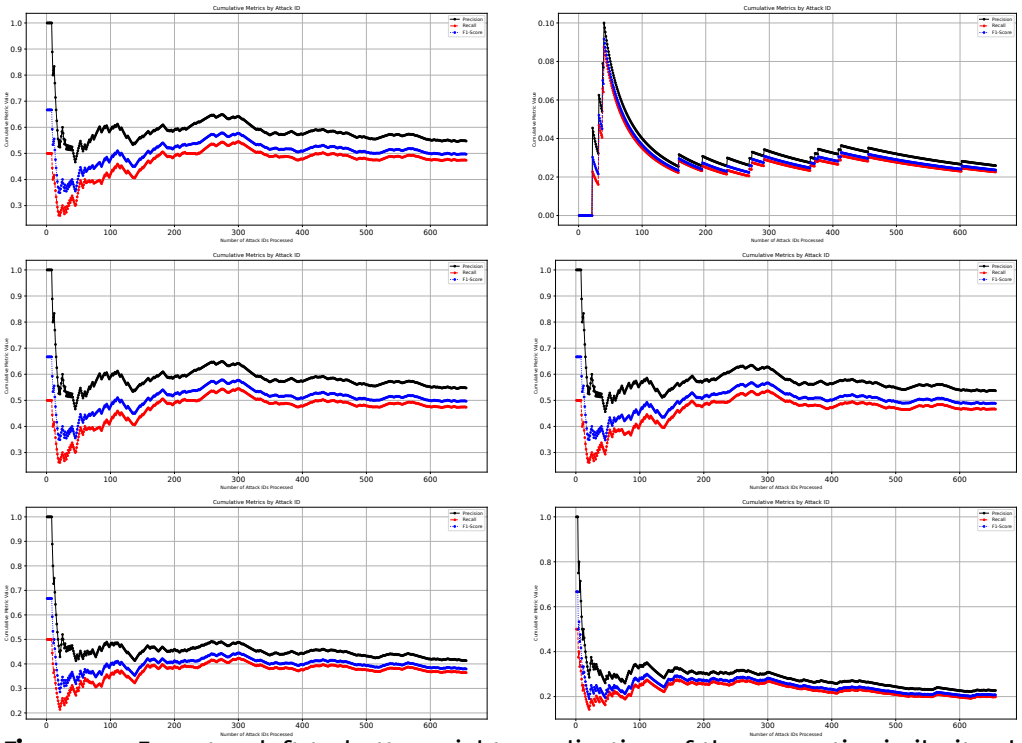


Figure 7.4: From top-left to bottom-right: application of the semantic similarity algorithms in the static case (6 scenarios), and cumulative calculation of the machine learning metrics

Recursive Case Considering the recursive case, illustrated in Figure 7.5, it is primarily different from the static case, with the only equality obtained in the application of the AMax1 algorithm with 2 models. Compared to the static case, there is a reversal of direction, where the recall curve takes over; therefore, the number of true positives increases significantly. The growth curve increases until just before the 300 attacks considered, and then stabilises in values that we could consider satisfactory, between 0.65 and 0.60, in the case of AMax1 with 2 models, 0.60 and 0.55 in the case of RMax3 with 2 models and RMax3 with 3 models. The recall and F1-score obtain the same growth and stabilisation trend, but with significantly lower values, reaching at most 0.50 (F1-score in AMax1 with 2 models) and a value between 0.40 and 0.34 for precision in all cases. So the considered algorithms obtain a good amount of true positives, but at the same time generate a quantity of false positives that influences the precision.

Compared to the static case, even in the recursive case, the performance of the RMax3 algorithm with 3 models and RMax2 with 4 models is relatively stable; that is, the differences in the values between the vari-

ous metrics are even more reduced to be almost identical. Respectively, the values stabilise in the same intervals as the static case.

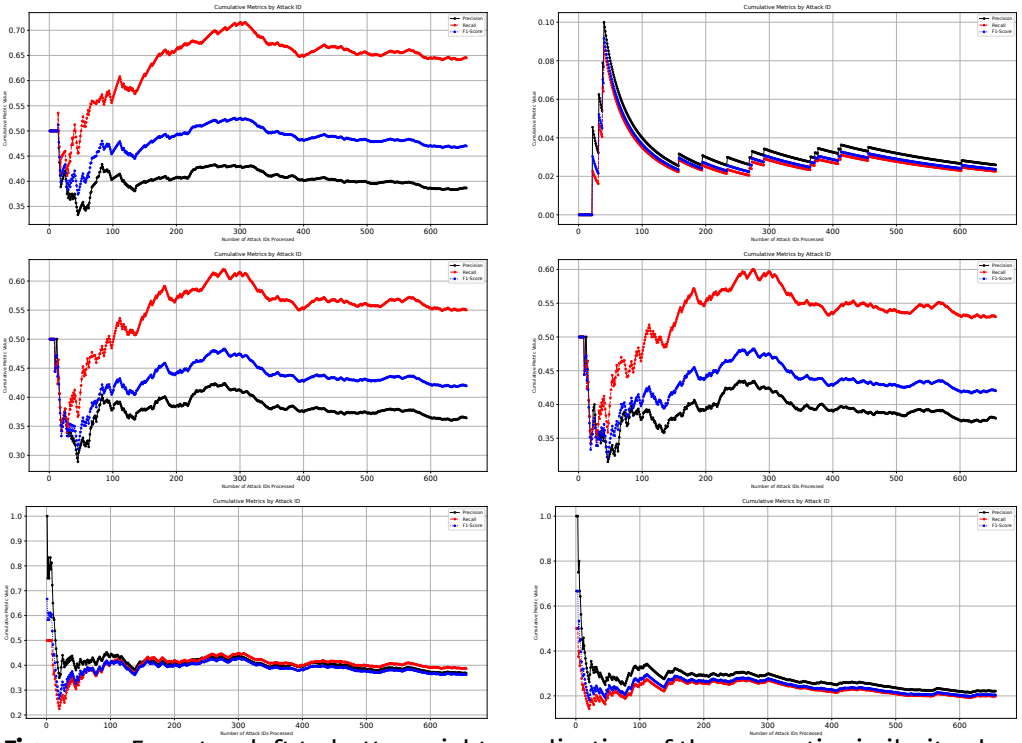


Figure 7.5: From top-left to bottom-right: application of the semantic similarity algorithms in the recursive case (6 scenarios), and cumulative calculation of the machine learning metrics

7.4.3 Non cumulative plots

In this last subsection, we investigate the behaviour of the metrics, individually for each technique, i.e. in a non-cumulative manner. This benchmarking allows us to understand how well semantic similarity performs in the individual technique, without showing the general trend (already done in the previous section).

Static Case Considering the static case, illustrated in Figure 7.6, it can be noticed, with a closer perspective, what has already been depicted in the histograms. That is, the most crucial information that can be obtained is that there is a balance between techniques that get, in terms of metrics, values close to zero or values close to one, with some verifiable exceptions between about 0.6 and about 0.45. This indicates that there is a balance between tactics that are correctly identified, tactics that are

incorrectly identified and tactics that are not recognised. This trend, in this not cumulative static case, occurs in almost all scenarios, starting with the application of the AMax1 algorithm with 2 models, where an imbalance of the metrics in favour of values close to zero is quite evident, and RMax3 with 3 and RMax2 with 4 models, where there is a similar imbalance, but less clear, that is, the values of the metrics close to 1 are still present in many attack instances. Furthermore, it can be noted that there is a pattern in which this behaviour of the metrics can occur. There are sequences of techniques of varying lengths in which the values are dense, followed by small voids. This indicates that in some consecutive techniques, the semantic similarity algorithms behave in the same way, i.e. they produce the same results in terms of metrics.

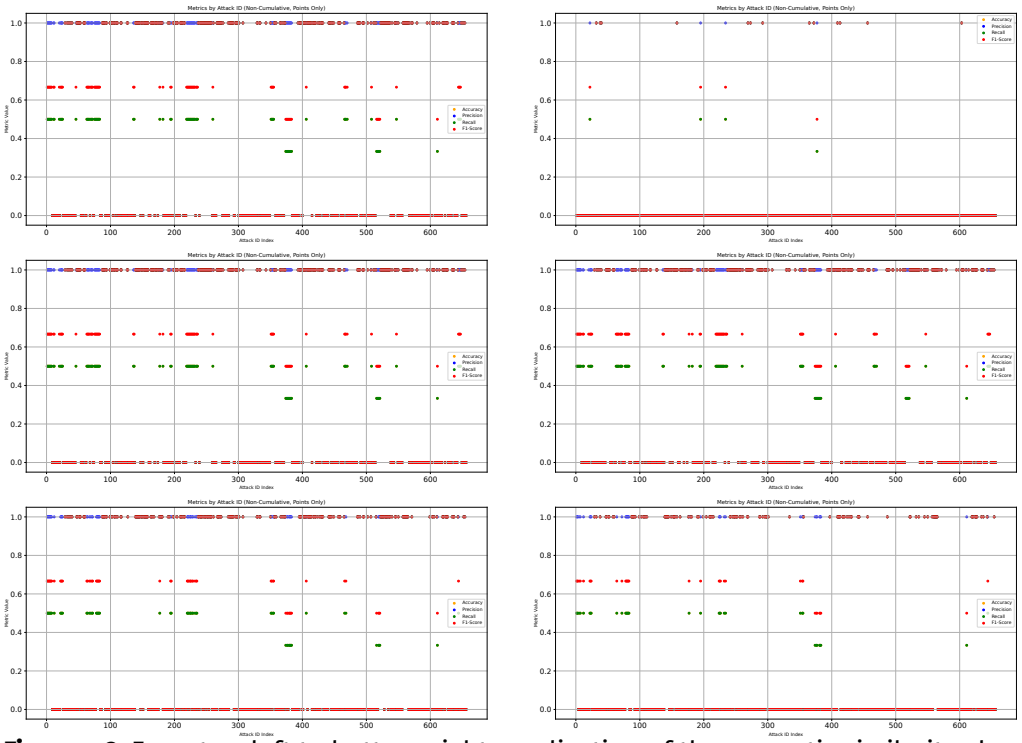


Figure 7.6: From top-left to bottom-right: application of the semantic similarity algorithms in the static case (6 scenarios), and calculation of the machine learning metrics for each ATT&CK ID (non cumulative)

Recursive Case In this recursive case, illustrated in Figure 7.7, compared to the static case, it can be noted that the metrics are more frequent in medium intervals, generally between 0.4 and 0.8, in almost all cases, except the application of the AMax1 algorithm with 2 models where the imbalance in favour of values close to 0 is confirmed. The recursion,

therefore, dampens the net values of the metrics of 0 and 1, or rather, the recursion causes a less marked difference between the resulting values of true positives, false positives or false negatives. Even in this non-cumulative recursive case, it is possible to notice a pattern, consistent with the static case, considering the techniques, with dense regions that are alternated by shorter regions without values.

Figure 7.7: From top-left to bottom-right: application of the semantic similarity algorithms in the recursive case (6 scenarios), and calculation of the machine learning metrics for each ATT&CK ID (non cumulative)

7.5 Benchmarking `Setàd`: Additional Considerations

In this section, we focus on two specific aspects. The first concerns the accuracy metrics, i.e. how accuracy differs from the metrics already presented in the previous graphs. The second concerns which models emerged as *leaders*, i.e. those that most frequently achieved the highest semantic similarity values — in this case, for convenience, we limit our benchmarking to applying the AMax algorithm with a single model.

7.5.1 Special focus on accuracy

We dedicate a special focus to the accuracy metric because in previous graphs, it can be observed that there was no room for accuracy, i.e. accuracy is not visible as a distinct curve. This is because accuracy values often overlap or closely align with those of other metrics, such as precision, recall, or F1-score.

This behaviour can be attributed to the specific characteristics of ATT&CK, where the inherent distribution of techniques makes it balanced. Under such an assumption, accuracy assumes values similar to those of the other metrics, since the number of true positives and true negatives contributes uniformly to the overall evaluation of the model. For this reason, we provide a separate illustration of accuracy curves in Figure 7.8.

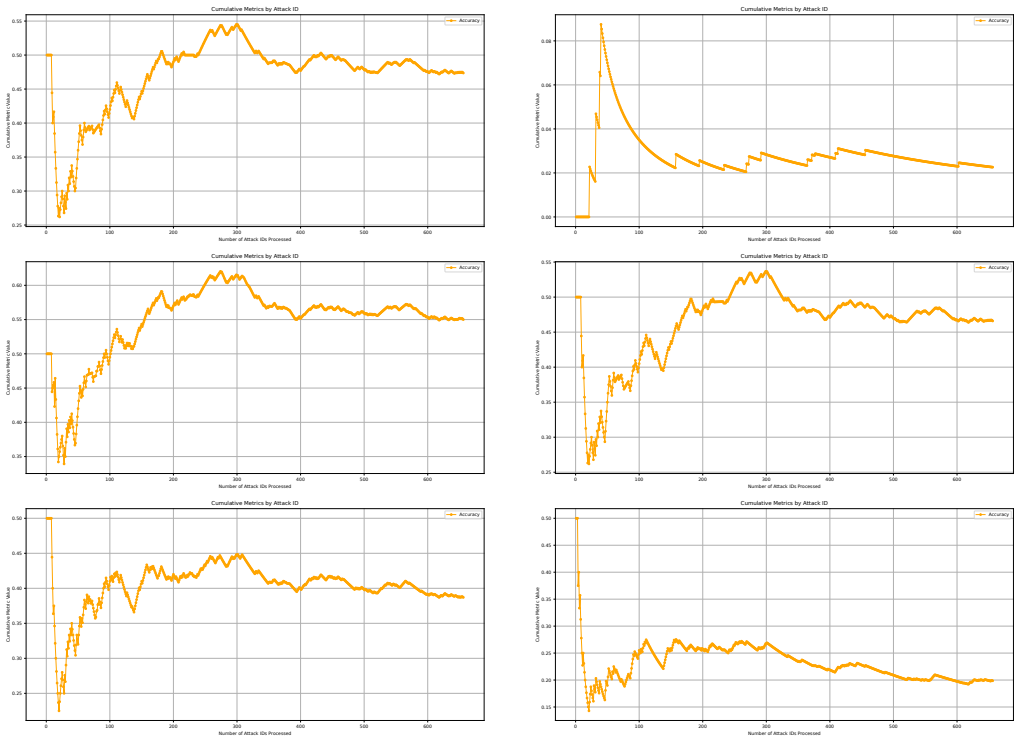


Figure 7.8: From top-left to bottom-right: application of the semantic similarity algorithms in the recursive case (6 scenarios), and cumulative calculation of only accuracy

Understanding which metric accuracy overlaps with is not always apparent, especially in cases where the curves of other metrics are themselves almost overlapping. Although it is evident that the accuracy resulting from applying AMax1 with 1 model is overlapping with pre-

cision, this distinction is less clear when applying RMax3 with 3 and RMax2 with 4 models, where precision, recall and F1-score are almost overlapping. Of course, an empirical evaluation of the results obtained, rather than the curves, would allow more precise conclusions to be drawn.

7.5.2 Special focus on models

This special focus investigates which models acted as *leaders*, i.e. those models that most frequently produced the highest semantic similarity scores when correlating a technique with a tactic. Unlike the previous experiments, which explored model agreement across a minimal number of models, this benchmarking evaluates each model in isolation; hence, the model that obtains the highest value of semantic similarity. This setup allows us to identify which models most often achieved the top semantic similarity values, thereby revealing the models that most strongly influenced the correlation results.

By calculating the accuracy even in this setting as well (Figure 7.9), a similar trend to those previously analysed can be noted. This outcome is expected since, in previous cases, considering a minimum set of models for convergence can lead to both correct mappings and incorrect mappings. The accuracy reported in this special setting shows that this proportion remains relatively balanced. Naturally, higher accuracy values are obtained in the recursive case than in the static case.

Further, Figure 7.10 shows which were the leading models, i.e., whose semantic similarity value was chosen the most, because it was the highest among the others.

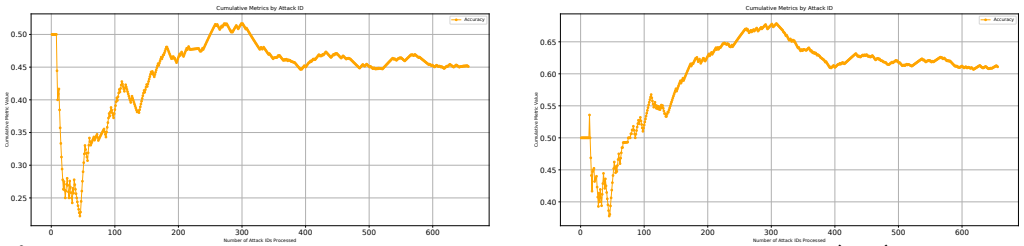


Figure 7.9: Accuracy reached by the single models, in the static case (left) and in the recursive case (right)

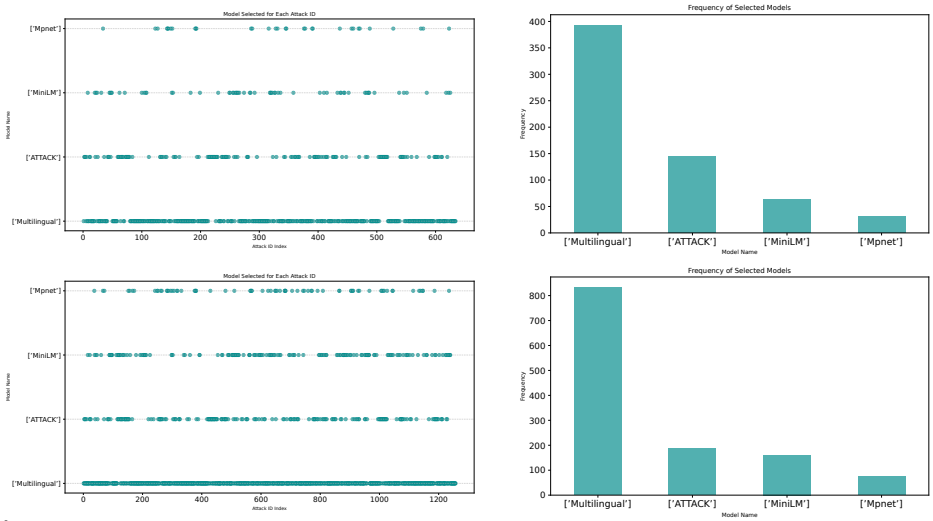


Figure 7.10: Models from which the highest value for technical-tactical correlation was derived. The static case is at the top, the recursive case at the bottom. On the left, the models considered in the related techniques, on the right, the number of models.

From this benchmarking, we can deduce that, although we are in the security domain, the only security-oriented model was not found to be the leader, and it still resulted in the second most frequent. Instead, the Multilingual model emerges as the leader, with a wide lead over the others. Considering the general accuracy trend, which is not excessively high, we can conclude that even high semantic similarity values may not fully capture the correctness of a correlation.

7.6 Automating `Setàd`

`Setàd` inherits the peculiarities of the semantic similarity step of `WISARD`. Naturally, the formalisation of `Setàd` is manually performed, while its correctness is automatically proven. Benchmarking `Setàd` can be considered fully automatic because, as in `MOSKAD`, we leveraged the ATT&CK framework as a ground truth. Hence, the discussed performance metrics are automatically calculated.

7.7 Concluding Remarks

Modern cyber attacks have taken on a collective connotation. Thanks to the various taxonomies made available by the scientific community, it is possible to touch and analyse the primary and most common attack

techniques used by the most well-known groups. This is the case of the ATT&CK framework, which inherently is a ground truth. The existing division into techniques and tactics allows us to test different methods to verify their effectiveness in deriving tactics related to specific techniques.

For this, we introduced Setàd as a methodology for formalising and benchmarking the performance of the families of algorithms of the semantic similarity step presented in WISARD, applied to the ATT&CK framework.

From this benchmarking, we can generally say that applying semantic similarity can be a promising solution, though not definitive, to identify the tactics from the techniques in ATT&CK. The histograms suggest a bifurcated distribution of the metric values: a substantial portion of the values are concentrated around one, indicating a high degree of confidence in specific associations; equally, there is a balance with a similar quantity of metrics that have values close to zero, indicating a failure by the semantic similarity to capture the correct mappings. This inherent limitation suggests that the semantic similarity could perform well in obtaining certain relationships, but it could perform poorly in obtaining others.

The key takeaway from applying Setàd is that using semantic similarity to address the problem correctly should be complemented by additional validation methods or strategies to ensure more complete and reliable results. as already shown in WISARD.

Chapter 8

Seer: Bridging Compliance with Security Directives and Vulnerability Assessment through Machine Learning and Fuzzing

Every new beginning comes from some other beginning's end.

— Seneca

This chapter is based on the publication "*Poster: Machine Learning for Vulnerability Detection as Target Oracle in Automated Fuzz Driver Generation*" [28].

This chapter illustrates Seer, a hybrid methodology composed of machine learning and fuzzing. The target of Seer is to identify a likely vulnerable function affected by one or some CWEs by leveraging machine learning as a *target oracle*, automatically generating *fuzz drivers*, and in the end fuzzing the target function to find bugs which could confirm the vulnerability predicted by the target oracle. We apply Seer on an existing vulnerability in libgd, with a plan for large-scale evaluation.

Seer represents the final methodology that allows vulnerability assessment activities to be carried out, starting from non-compliance with the security measures of the directives.

8.1 Introduction

The motivation for taking this further and final step in the present dissertation is dictated by the scope that the NIS 2 Directive can achieve. Through WISARD, we have linked security measures to attack patterns, which do not necessarily involve software attacks but may concern network attacks or social engineering, among others. Nevertheless, the security measures in the Directive may also refer to implementation-specific aspects, particularly those related to software development. For this reason, we considered the CWEs derived through WISARD as an additional resource for further investigating the effects of non-compliance, this time considering the lower-level aspects, technical facets of a com-

pany, i.e. the software it develops or uses. The rationale behind this final contribution – which, it is worth mentioning, is in its early stages and, compared to previous methodologies, requires further investigation in future research – is to attempt to predict whether specific functions, either internally developed or externally sourced, are potentially vulnerable to one or more CWEs derived from WISARD.

In particular, we investigate whether machine learning for vulnerability detection may result in an effective target oracle for automated fuzz driver generation, and to what extent such a combined method confirms the true positives, and/or reduces the number of false positives. We present the design and workflow that combines the two techniques. We validate Seer by selecting a confirmed vulnerable function from the DiverseVul dataset and successfully applying OSS-Fuzz-Gen to generate a fuzz driver that triggers the vulnerability. The target function originates from a project already included in the OSS-Fuzz infrastructure[1], ensuring compatibility with OSS-Fuzz-Gen, but is not currently covered by an existing fuzz target. This allows us to generate a novel fuzz driver and achieve previously unreached code coverage.

8.2 Related Work

The use of deep learning for vulnerability classification and guided fuzzing has been previously explored by Zhu et al. [147]. Their approach does not fuzz the function identified as vulnerable directly. Instead, they use directed fuzzing to generate inputs that reach specific code fragments within the vulnerable function. This results in a focus on vulnerable segments of the function, rather than the function as a whole. In contrast, our work adopts a different strategy. By applying the concept of harnessing, we are able to fuzz the entire function identified as potentially vulnerable, treating it as a standalone unit. This difference has important implications. As Zhu et al. acknowledge, computing reachability and distance to vulnerable code fragments introduces limitations. In the worst-case scenario, the fuzzer might fail to reach the predicted vulnerable region altogether. Furthermore, if the model identifies only a fragment rather than the full function, the effectiveness of directed fuzzing may be compromised, especially since predicting precisely the vulnerable code portion is arguably more complex than predicting vulnerability at the function level. By abstracting the target to

[1] https://github.com/google/oss-fuzz

the full function, our approach avoids the reachability issue entirely. The fuzzing is applied directly to the harnessed function, enabling a broader and more robust exploration surface without relying on distance-guided heuristics.

Risse et al. [114] have shown that top-performing ML4VD models are unable to distinguish between functions that contain a vulnerability and functions where the vulnerability is patched. Consequently, without a definitive solution, we expect an increase in false positives over time, which could be mitigated by `Seer`. The state-of-the-art employs *Directed Greybox Fuzzing (DGF)* to steer the generation of inputs that can reach a specific program location. Zhu et al. [147], Yu et al. [143], already employed ML4VD as a target oracle for DFG. However, DFG itself does not ensure that the target function can be reached. For example, a compiled binary could never call a library function. Our work employs automated fuzz driver generation to generate fuzz drivers which call the target function.

8.3 Special Focus On Machine Learning And Fuzzing

In recent years, two automated techniques have emerged for finding 0-day vulnerabilities in functions: Machine Learning for Vulnerability Detection (ML4VD) [114] and *Fuzzing* for vulnerability discovery.

ML4VD models, when trained on large datasets, are employed to determine whether a given set of functions may exhibit specific vulnerabilities. However, the method involves static analysis, which cannot verify the vulnerability at runtime, may suffer from a significant false-positive rate [31], and ultimately, top-performing models may not be able to differentiate between vulnerable functions and patched functions [114].

On the other hand, fuzzing employs dynamic analysis, reducing false positives from static analysis. However, no push-button fuzzing technique exists, as a *fuzzer* requires a *fuzz driver*, a test harness for parsing inputs and invoking the target function that, in turn, requires deep knowledge about the target function and the corresponding library and extensive manual work [15]. *Automated Fuzz Driver Generation (AFDG)*, despite its challenges, solves the burden, with OSS-Fuzz-Gen [83] standing on top due to the use of Large Language Models (LLMs). Contextually, a library could include several functions, and *target oracles* are employed to identify *interesting* functions [138], i.e. functions determined to be likely interesting targets. For example, OSS-Fuzz-Gen prioritises functions

relying on Fuzz Introspector's[2] heuristics, which mainly consider the cyclomatic complexity of the target function or the simplicity to generate the fuzz driver. Nevertheless, these heuristics do not account for the likelihood of a function being vulnerable. Hence, we propose `Seer`, a combined methodology that employs ML4VD models as a target oracle to prioritise relevant functions for the AFDG.

8.4 The `Seer` Methodology

Nowadays, the most effective way to predict features in a particular object is through the application of machine learning. However, such predictions are even more accurate and efficient if the models used for a certain task, for example, the prediction of CWEs, have been trained on datasets specific to that task and are as detailed as possible. In our case, the use of machine learning is combined with the `DIVERSEVUL` dataset, which provides one of the most comprehensive collections of vulnerable functions annotated with zero or more CWEs. The core idea is to train high-performance models on `DIVERSEVUL`, input the functions under analysis, predict whether these functions are susceptible to any CWEs, and select the functions that have been classified as having CWEs from the set derived from the attack patterns obtained with `WISARD`.

Subsequently, the remaining steps of the `Seer` methodology are applied, completing the proposed methodological flow from regulatory directives to technical vulnerability validation.

The design of `Seer` employs two main techniques: an ML4VD model as the vulnerability detection component (static analysis of the target function code), and AFDG, for ultimately applying fuzzing as the discovery component (dynamic analysis that uncovers vulnerabilities during execution of the target function).

The overall workflow of `Seer`, illustrated in Figure 8.1, comprises three steps. It represents a generalised pipeline for AFDG and emphasises the main contribution of this work. Namely, the use of the ML4VD model as a target oracle.

[2] https://github.com/ossf/fuzz-introspector

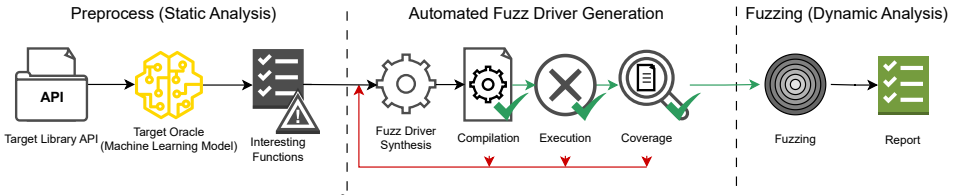


Figure 8.1: Workflow of `Seer`

Preprocess (Static Analysis). Our methodology, `Seer`, first identifies functions in the target library API that are likely to be vulnerable. The objective is accomplished by an ML4VD model, which performs a static analysis of each function and flags those that may present one or more weaknesses according to the *CWE* framework by MITRE, hence obtained by `WISARD`. Ultimately, the flagged functions are considered potentially interesting functions for further analysis by fuzzing.

Automated Fuzz Driver Generation. Subsequently, the interesting functions are selected for the AFDG process to begin. This is an iterative process in which the fuzz driver synthesis produces a candidate fuzz driver. The candidate must (1) compile, (2) execute without immediate logical failure, and (3) gather sufficient coverage to ensure the input is correctly injected. Once these steps are met, the candidate proceeds to the next stage of the workflow.

Fuzzing (Dynamic Analysis). At this stage, the fuzzing process begins, leveraging the previously generated fuzz driver to inject the target function with a large volume of malformed inputs. At the expiration of the time budget, the fuzzer either discovers a crashing input that confirms the vulnerability or does not.

8.5 Case Study: LIBGD Library

Target Selection. To evaluate the design of `Seer`, we conducted initial experiments on the LIBGD[3] library. In particular, we assumed to have already a ML4VD model trained on the DiverseVul dataset [31], which is considered the best collection of vulnerable functions in C/C++.

Subsequently, we selected among the projects in DiverseVul one already included in the OSS-Fuzz infrastructure and with Fuzz Introspector reports available. We used such requirements to identify functions

[3] https://github.com/libgd/libgd

currently not covered by existing fuzz drivers in such a project. Consequently, we directly take an uncovered function labelled with CWEs as most likely classified as vulnerable, belonging to the dataset itself.

In particular, we chose the function *gdImageWebpPtr*, labelled as weak to **CWE-415: Double Free**[4], which is confirmed by **CVE-2016-6912**[5].

Experimental Setting. Subsequently, we employed OSS-Fuzz-Gen to generate a fuzz driver. Initially, OSS-Fuzz-Gen relies on Fuzz Introspector to retrieve the target function signature and its corresponding arguments, which are provided in YAML format, as illustrated in Figure 8.2. Since the target function is selected by the target oracle, we assume that our ML4VD model has flagged the *gdImageWebpPtr* function as potentially vulnerable.

```yaml
functions:
    - name: "gdImageWebpPtr"
params:
    - name: "im"
      type: "gdImagePtr"
    - name: "size"
      type: "int *"
return_type: "void *"
signature: "BGD_DECLARE(void *) gdImageWebpPtr (gdImagePtr im, int *size)"
language: "c++"
project: "libgd"
target_name: "gd_webp_fuzzer"
target_path: "gd_webp_fuzzer.cc"
```

Figure 8.2: Function Signature of *gdImageWebpPtr*

After this minimal setup, the AFDG process starts with the creation of a prompt from a template to be provided to a Large Language Model (LLM). The prompt template is shown in Figure 8.5. In particular, we used the standard prompt template provided in the official repository of OSS-Fuzz-Gen; the only difference is the embedded information about the target's weaknesses we added in the prompt, which is shown in red.

[4] https://cwe.mitre.org/data/definitions/415.html
[5] https://nvd.nist.gov/vuln/detail/CVE-2016-6912

OSS-Fuzz-Gen Prompt

1 - System Prompt

You are a security testing engineer who wants to write a C++ program to discover memory corruption vulnerabilities in a given function-under-test [...]

2 - C++ Specific Instructions

Use <code>FuzzedDataProvider</code> to generate these inputs [...]

3 - Instructions and Examples

[...] Do not create new variables with the same names as existing variables. WRONG: <code>
int LLVMFuzzerTestOneInput(const uint8_t *data, size_t size) {
void* data = Foo();
} </code> [...]

4 - Problem Statement

Your goal is to write a fuzzing harness for the provided function-under-test signature [...] <function signature>
Note: The function is a candidate for the vulnerability CWE-415 (Double Free)

Figure 8.3: OSS-Fuzz-Gen prompt

In our initial experiments, we employed *GPT-4* with a temperature setting of 0. Although the model successfully generated valid fuzz drivers on first attempts, the fuzz drivers struggled to find a bug and ultimately confirm the vulnerability. This is because the specific vulnerability can be detected whether *gdImageWebpPtr* is invoked by passing a sufficiently large image. Instead, the fuzz drivers presented a hard-coded limit cap on the size of the image, likely to prevent out-of-memory errors during the fuzzing campaign. To solve the issue, we instructed the model to allow large images without setting a cap. From the initial experiments, we can conclude that if a valid fuzz driver fails to find a bug, this does not rule out a vulnerability, and further instructions based on the expected vulnerability could be needed.

Ultimately, our contribution aims to (1) prioritise functions likely to expose a weakness, (2) confirm a vulnerability whenever a critical bug is found.

8.6 Automating `Seer`

The `Seer` methodology contains automatic and manual components for both the static and dynamic analysis cases. Within the static analysis, the prediction of the likely vulnerable function employs elements automatic at their core, such as the obtained CWEs from `WISARD` and the machine learning model. However, it may remain for human scrutiny to decide which functions (or set of) should be analysed by the model.

At the same time, within the dynamic analysis, although the fuzzing component is fully automatic, it initially relies on crafting accurate prompts for generating correct fuzzing harnesses. Hence, although each harness is automatically created, human intervention is necessary for guiding its creation, since wrong ones can be created.

8.7 Concluding Remarks

`Seer` represents an innovative methodology that combines the potential of the two main vulnerability research methods, namely, machine learning and fuzzing. This combination leverages the integration of the potential of both methods to achieve an approach that is potentially even more accurate. Furthermore, this combination allows us to propose a new heuristic, while Fuzz Introspector uses two heuristics to determine functions likely to be interesting targets. However, these heuristics fall short of considering only functions having high cyclomatic complexity and containing *parse* in their name (Heuristic 1), and accepting the same argument types as the fuzzing interface *LLVMFuzzerTestOneInput* (Heuristic 2). `Seer` proposes a third heuristic, in which an ML4VD model identifies potentially vulnerable functions.

However, regarding the current study, potential *threats to validity* may exist. One threat may be the following: as the case study involves a CVE from 2015, this may raise a question about whether the code to reproduce such vulnerability is memorised by the model from the training data. Future studies will focus on evaluating the effectiveness of novel candidate functions. Hence, future work will primarily focus on applying `Seer` to a broader range of functions. Our plan encompasses

the integration of an ML4VD model as a target oracle (third heuristic) into Fuzz Introspector and evaluating its effectiveness on at least ten projects from both OSS-Fuzz and DiverseVul, relying on OSS-Fuzz-Gen for AFDG.

Therefore, we plan to evaluate our target oracle in novel interesting functions, i.e. not included in the dataset. Realistically, functions developed by a company that is non-compliant with specific security measures, and from which potential CWEs have been derived through the application of the WISARD methodology. We will focus on functions in a test set, measuring the precision of both the target oracle and, ultimately, the AFDG process.

Chapter 9

Concluding the Dissertation

Ora quei giorni sono distanti e i
miei occhi pronti
Per dare forma ai nuovi orizzonti.

– Neffa & Deda

Security compliance constitutes a significant source of concern for many corporate decision-makers due to its complexity and cost. With the advent of security legislation, ensuring compliance is further complicated by the language style in which security legislation is written, mainly causing concern about translating the outlined security measures into concrete operational procedures. We provocatively hypothesise that the existence of the phenomenon of *non-compliance* leads to a broader condition of *insecurity*. Insecurity can be interpreted in multiple dimensions: *financial insecurity*, resulting from regulatory fines; *cyber-insecurity*, due to technological gaps and daily discovery of vulnerabilities. However, going beyond mere technological implications, non-compliance may also trigger, but is not limited to, *reputational insecurity*, stemming from image damage as a consequence of negative external perception, and *organisational insecurity*, due to lack of internal coherence, weak governance.

In this dissertation, we investigate this multifaceted relationship by analysing current security legislation and designing novel methodologies to bridge the gap between security legislation and security tasks. We specifically dedicated our focus to the NIS 2 Directive, as the newest European security directive.

This PhD study demonstrates how it is possible to reach low-level and specific actions and activities starting from high-level and abstract knowledge. We begin by analysing the content of the NIS 2 Directive *verbatim*, and terminate by illustrating a methodology for assessing vulnerabilities. These seemingly distant fields are sequentially connected by intermediate methodologies that step-by-step reduce the level of abstraction, making security activities increasingly specific.

9.1 Contributions And Implications

Our main contributions strictly rely on the methodologies thoroughly discussed in this dissertation.

We first contributed with SecOnto as a general methodology for the structured design of security directives. SecOnto represents the first step in refining security directives by representing them in an ontological format rather than in a textual and poorly structured form. This structured representation of the directive, according to criteria provided by SecOnto itself, supports interoperability for both compliance verification and attack detection. By applying SecOnto, we obtained NIS2Onto, a comprehensive representation of the NIS 2 Directive. NIS2Onto provides all agents and related security measures outlined in the directive with an ontological structuring.

The transition towards offensive activities is played by the WISARD methodology. Starting from the security measures outlined in the NIS 2 Directive, WISARD aims to identify a specific category of related attacks, namely attack patterns. The idea is that non-compliance with specific measures, especially technical measures rather than organisational measures, leads to specific attack patterns. Therefore, an adversary or security tester would exploit this regulatory gap for targeted offensive security activities. In taxonomic terms, the attack patterns represent the most general category and least detailed of attacks, hence the most suitable to be correlated with the very general measures of the NIS 2 Directive.

Although WISARD provides a knowledge base of attack patterns, the subsequent step introduced by the MOSKAD methodology was to structure these attack patterns sequentially and systematically for composing entire offensive killchains. With MOSKAD, the previously identified attack patterns are operationalised by being placed in the most appropriate steps, whose union produces entire attack killchains.

The main advantage of working with attack patterns is the close correlation they have with other taxonomies. By hierarchical definition, an attack pattern can be specialised in multiple CWEs, and a CWE can be specialised in numerous CVEs. We contributed with Seer as the last effort to reduce the gap between security legislation and security activities, and, in particular, vulnerability assessment activities. Seer uses the CWEs derived from WISARD to guide a hybrid approach that combines machine learning and fuzzing. CWEs can be used by machine learning to predict the predisposition of software to be affected by CWEs themselves. Through fuzzing, it is confirmed whether the functions expected to be

affected by CWEs are actually vulnerable, potentially confirming existing CVEs. Although the methodology has been detailed and tested on a use case, among the other methods, it is expected to be the one with the widest margin for research, being in its initial stage of investigation.

With `GTCheck` and `Setàd`, we provided an investigative overview of the application of different tools and algorithms in correctly obtaining, respectively, part of speech in security directives, and tactics from techniques in security frameworks.

9.2 Comparison With Existing Frameworks

Throughout the dissertation no specific frameworks have been analysed or specifically discussed. In particular, we refer to such *off-the-shelf* security frameworks employed within the modern security tasks for assessing vulnerabilities, compliance with specific standards, etc. Rather, for each methodological approach, we just limited ourselves to providing the most classical state of the art for comparing the idea behind it.

This section of comparison appears only in the conclusions because only at this point is the reader aware of the strategies, hence methodologies, presented within this dissertation. By consequence, the reader is aware of the full context and peculiarities that differentiate our proposed methodologies from the most well-known security frameworks. The research questions we thoroughly analyse within this dissertation differ substantially from the foundational criteria on which the modern security frameworks are built. Therefore, we briefly mention and compare some security frameworks that summarise the state of the art and are close to the overall strategy illustrated in this dissertation.

Standards such as ISOs, NIST, etc., are the primary focus of the modern security frameworks when offering the capability of compliance verification. The relative simplicity of interpreting the controls defined therein and the simplicity of mapping such controls into vulnerabilities, for example, allow for "easy" verification that a given software may be non-compliant with a specific control. Among others, OpenSCAP [2] for compliance verification offers open-source tools for auditing systems in relation to compliance standards and security baselines. Enterprise vulnerability scanners like Nessus, Qualys, and Rapid7 InsightVM [6, 4, 5] incorporate similar open frameworks and have integrated policy compliance modules that translate results to regulatory standards that, we remark, can be considered technical enough to easily guide the security operator.

Attack derivation, instead, can be considered similar but not necessarily close to the field of threat modelling. The concept of *non-compliance* is employed by a specific category of the LINDDUN framework [35], although LINDDUN operates in the context of privacy. Organisations can map threats with the aid of tools such as the OWASP Threat Dragon and Microsoft Threat Modelling Tool [1, 3]. Threat modelling, hence such tools, tackles the "why" of necessary controls, establishes which assets may be affected by specific threats, and risk assessment may be carried out in the end.

Our approach to compliance verification and attack derivation presents an innovative perspective, substantially different from the previous approaches. The first objective was to reduce the difficulties in operationalising the security measures, coming from a legal background, by providing structured and machine-readable representations, which proved useful also for attack derivation. The second objective was to *exploit* the possible non-compliance as an initial point to start targeted offensive security activities, where the single mandated security measures serve as input for active attack research.

Overall, we argue that our work proposes a genuine paradigm shift.

9.3 Analysis On Limitations

Directive-related Limitations One of the main limitations of this PhD dissertation lies in the applicability of the designed methodologies to industrial and fully operational contexts. Although each methodology provides real use cases, employs real security frameworks and well-known instruments and tools, the complete transposition of the methodologies to real-world use cases remains future work. This is due to the *age* of the NIS 2 Directive, which is particularly short. Even if promulgated in the last months of 2022, almost three years from now, its transposition to national security and implementation across Member States is still an ongoing process. As a consequence, a complete evaluation of the research questions provided within this dissertation remains limited to the scientific settings, rather than being empirically validated in operational environments. Nevertheless, with our research, we propose methodologies that go well beyond the mere transposition of the Directive, looking ahead to scenarios in which the Directive will have to be implemented and, reasonably, not complied with by some entities.

Methodologies-related Limitations Although the various methodologies have enabled us to answer the research questions posed fully, it would be reasonable and conceivable that improvements could be made, and therefore, that there may be some limitations.

`SecOnto` is a methodology that essentially provides a design to support compliance checking. We have carefully designed `SecOnto` to simplify the interoperable structuring of the directive as much as possible and make it easy to use by various entities. `SecOnto` breaks down complex elements into basic sub-elements and then recombines them with a structurally more queryable design than the one provided at the outset. However, possible limitations could arise from the interpretive flexibility of legal texts. Organisations may develop alternative interpretations of the original directive that diverge from the structure imposed by `SecOnto`.

The limitations of Wisard essentially lie in the possible limitations of the semantic methods and the responses of security experts. These limitations could, in particular, produce false negatives, i.e. those possible attacks that are not considered to be correlated. The following reasons may cause the limitations we hypothesise: *Limitations due to semantic similarity and related algorithms.* Algorithms with semantic similarity, and semantic similarity per se, may not be entirely effective in capturing all possible correlations. This may be due to the limitations of the models and the conditions of the algorithms, which, although designed to offer a degree of convergence, may be stringent. *Limitations due to ontological semantics and related prompt engineering.* Although we do not see any criticalities from an ontological point of view, since query execution and thus the retrieval of security measures is a deterministic process, limitations may arise in the invocation of the model. Even though it was invoked several times, to obtain answers that were numerically greater and, above all, consistent, a substantial disparity between the number of correlations found to semantic similarity can be noted. Thus, either the semantic similarity approach produces many false positives or the semantic ontology approach produces many false negatives. The final choice of intersecting the results of both, together with the validation base, aimed to solve this possible problem. *Limitations due to the involvement of the security experts.* Although the questionnaires were purposely designed to simplify the filling-in process, remove ambiguities and make the process as smooth as possible, limitations purely concerning the human sphere may arise. In particular, that security expert's responses are not entirely reliable, hence concerns arising from the filling in of questionnaires. This is mainly due to the process, which may have been perceived as tedious by them. Again, the idea of using the results of this activity for the intersection

step rather than as ground truth is intended to limit the possibility of such issues.

MOSKAD defines a flow that begins with attack patterns and ends with killchain steps. To facilitate this transition, we considered it necessary and appropriate to use ATT&CK as a bridge to any killchain, given its status as a killchain itself. However, relying on this intermediary mapping process could have certain drawbacks. Any errors or restrictions in the mapping process, whether brought on by the framework's structure or the tools and techniques employed, could compromise the accuracy and comprehensiveness of the final killchain representation because ATT&CK serves as a link between the abstract attack patterns and the concrete killchain phases.

As previously noted, Seer is in the early stages of development and necessitates future research. Nonetheless, some limitations of the proposed approach are strictly related to its individual components. Both fuzzing and machine learning have well-established drawbacks in the literature. For example, fuzzing may have issues with coverage and scalability, while machine learning approaches may have restricted generalisability, data reliance, and interpretability issues. Seer was explicitly developed to address some of these constraints by utilising a hybrid approach that combines the confirming capability of fuzzing with the predictive powers of machine learning. The efficacy of this combination, however, still needs further scientific support and is hence still up for debate.

9.4 Beyond The Dissertation

Each of the unique Key Exploitable Results that have been codified from the introduction and described throughout this dissertation represents a methodological or technological progress. Future extensions are made possible by these KERs, which serve as the cornerstone of the suggested research impact.

In particular, future work may be extensive and encompass a wide range of activities, including both scientific and industrial endeavours. From a scientific perspective, a natural continuation of this work is the further investigation of the proposed methodologies for potential refinements and enhancements. We see the possibility of refining the work to cover a broader spectrum of cases, both in terms of the regulatory and attacks landscape, and therefore defences, especially with the evolution of the mirror scenario concerning artificial intelligence.

Instead, from an industrial perspective, the proposed methodologies could be integrated into real-world security engineering workflows, as regulatory compliance and risk assessment are core requirements. Organisations could benefit from automation pipelines that translate high-level regulatory texts into actionable security tasks. The methodologies `SecOnto`, `WISARD`, `MOSKAD`, and `Seer` may be integrated into tools to support decision-makers and technical staff in aligning operational practices with evolving regulatory requirements. Such methodologies could be particularly valuable in highly regulated sectors, such as finance, healthcare, and critical infrastructure, where both non-compliance and security breaches carry significant risk, especially in rapidly changing regulatory contexts.

Appendix A

Chapter 2 Appendix

A.1 Article 11 Objects

Item 11.1

Manual

N11.1.1

N11.1.2.a a high level of availability of their communication channels by avoiding single points of failure, and shall have several means for being contacted and for contacting others at all times; they shall clearly specify the communication channels and make them known to constituency and cooperative partners

N11.1.2.b located at secure sites

N11.1.2.c with an appropriate system for managing and routing requests

N11.1.2.d the CSIRTs shall ensure the confidentiality and trustworthiness of their operations

N11.1.2.e availability of their services and their staff is trained appropriately

N11.1.2.f redundant systems and backup working space to ensure continuity of their service

ClausIE

S11.1.1

S11.1.2.a a high level of availability of their communication channels

S11.1.2.b NONE

S11.1.2.c NONE

S11.1.2.d the CSIRTs shall ensure the confidentiality and trustworthiness of their operations

S11.1.2.e availability of their services

S11.1.2.f NONE

Item 11.2

Manual

N11.2.1 that their CSIRTs jointly have the technical capabilities necessary to carry out the tasks referred to in paragraph 3

N11.2.2 that sufficient resources are allocated to their CSIRTs to ensure adequate staffing levels for the purpose of enabling the CSIRTs to develop their technical capabilities

ClausIE

S11.2.1 that their CSIRTs jointly have the technical capabilities necessary to carry out the tasks referred to in paragraph 3

S11.2.2 that sufficient resources are allocated to their CSIRTs to ensure adequate staffing levels for the purpose of enabling the CSIRTs to develop their technical capabilities

...

A.2 Article 23 Objects

Item 23.1

Manual

N23.1.1 that essential and important entities notify, without undue delay, its CSIRT or, where applicable, its competent authority in accordance with paragraph 4 of any incident that has a significant impact on the provision of their services as referred to in paragraph 3

N23.1.2 the recipients of their services of significant incidents that are likely to adversely affect the provision of those services

N23.1.3 that those entities report, inter alia, any information enabling the CSIRT or, where applicable, the competent authority to determine any cross-border impact of the incident

N23.1.4 the notifying entity to increased liability

ClausIE

S23.1.1 that essential and important entities notify, without undue delay, its CSIRT or, where applicable, its competent authority in accordance with paragraph 4 of any incident that has a significant impact on the provision of their services as referred to in paragraph 3

S23.1.2 the recipients of their services of significant incidents that are likely to adversely affect the provision of those services

S23.1.3 that those entities report, inter alia, any information enabling the CSIRT or, where applicable, the competent authority to determine any cross-border impact of the incident

S23.1.4 the notifying entity to increased liability

...

Item 23.8

Manual

N23.8.1 notifications received pursuant to paragraph 1 to the single points of contact of other affected Member States.

ClausIE

S23.8.1 NONE

Item 23.9

Manual

N23.9.1 a summary report, including anonymised and aggregated data on significant incidents, incidents, cyber threats and near misses notified in accordance with paragraph 1 of this Article and with Article 3

N23.9.2 technical guidance on the parameters of the information to be included in the summary report

N23.9.3 the Cooperation Group and the CSIRTs network about its findings on notifications received every six months

ClausIE

S23.9.1 a summary report, including anonymised and aggregated data on significant incidents, incidents, cyber threats and near misses notified in accordance with paragraph 1 of this Article and with Article 3

S23.9.2 technical guidance on the parameters of the information to be included in the summary report

S23.9.3 the Cooperation Group and the CSIRTs network about its findings on notifications received every six months

Item 23.10

Manual

N23.10.1 to the competent authorities under Directive (EU) 2022/2557 information about significant incidents, incidents, cyber threats and near misses notified in accordance with paragraph 1 of this Article and with Article 30 by entities identified as critical entities under Directive (EU) 2022/2557

ClausIE

S23.10.1 NONE

Item 23.11

Manual

N23.11.1 implementing acts further specifying the type of information, the format and the procedure of a notification submitted pursuant to paragraph 1 of this Article and to Article 30 and of a communication submitted pursuant to paragraph 2 of this Article

ClausIE

S23.11.1 NONE

Appendix B

Chapter 4 Appendix

Code B.1: SPARQL for Differential Analysis on specific article GDPR

```
SELECT ?article ?action ?object
   WHERE{
      nis:ImportantEntity-ExAnte owl:equivalentClass ?a .
      ?a owl:intersectionOf ?b .
      ?b rdf:rest* ?c .
      ?c rdf:first ?article .
      ?article owl:equivalentClass ?e.
      FILTER (?article = nis:Art21Par2-j-Entity)
      ?e owl:intersectionOf ?f .
      ?f rdf:rest* ?t .
      ?t rdf:first ?s .
      ?s owl:onProperty ?action .
      ?s owl:someValuesFrom ?object .
     MINUS {
      nis:CompliantOrganisation ?action ?objInd .
      ?objInd rdf:type ?object .
     }
   }
```

Code B.2: SPARQL for Specific Search on entities

```
SELECT DISTINCT ?specificClass
WHERE {
  ?anyClass owl:equivalentClass ?definition .

 {
    ?definition (owl:intersectionOf|owl:unionOf|rdf:rest*/rdf:first)*
        ?specificClass .
  }
  UNION
  {

    ?definition (owl:intersectionOf|owl:unionOf|rdf:rest*/rdf:first)*
        ?restriction .
    ?restriction owl:someValuesFrom ?specificClass .
  }
  FILTER regex (STR(?specificClass), "CSIRT")
}
```

Code B.3: SPARQL for Integration from GDPR

```
INSERT {
  nis:GDPR_Art6Par2 a owl:Class ;
                owl:equivalentClass nis:MemberState .
}
WHERE {
  #
}
```

References

[1] Microsoft Threat Modeling Tool. https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool.

[2] OpenSCAP Project. https://www.open-scap.org/.

[3] OWASP Threat Dragon. https://owasp.org/www-project-threat-dragon/.

[4] Qualys Cloud Platform. https://www.qualys.com/.

[5] Rapid7 InsightVM. https://www.rapid7.com/products/insightvm/.

[6] Tenable Nessus Professional. https://www.tenable.com/products/nessus.

[7] Basel Abdeen, Ehab Al-Shaer, Anoop Singhal, Latifur Khan, and Kevin Hamlen. Smet: Semantic mapping of cve to att&ck and its application to cybersecurity. In *Data and Applications Security and Privacy XXXVII: 37th Annual IFIP WG 11.3 Conference, DBSec 2023, Sophia-Antipolis, France, July 19–21, 2023, Proceedings*, page 243–260, Berlin, Heidelberg, 2023. Springer-Verlag.

[8] Ehsan Aghaei and Ehab Al-Shaer. Cve-driven attack technique prediction with semantic information extraction and a domain-specific language model. *ArXiv*, abs/2309.02785, 2023.

[9] Khandakar Ashrafi Akbar, Sadaf Md Halim, Yibo Hu, Anoop Singhal, Latifur Khan, and Bhavani Thuraisingham. Knowledge mining in cybersecurity: From attack to defense. In Shamik Sural and Haibing Lu, editors, *Data and Applications Security and Privacy XXXVI*, pages 110–122, Cham, 2022. Springer International Publishing.

[10] Bader Al-Sada, Alireza Sadighian, and Gabriele Oligeri. Mitre att&ck: State of the art and way forward. *ACM Comput. Surv.*, 57(1), October 2024.

[11] Mutlaq Alotaibi, Steven Furnell, and Nathan Clarke. Information security policies: A review of challenges and influencing factors. In *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 352–358, 2016.

[12] Masumi Arafune, Sidharth Rajalakshmi, Luigi Jaldon, Zahra Jadidi, Shantanu Pal, Ernest Foo, and Nagarajan Venkatachalam. Design and development of automated threat hunting in industrial control systems. In *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 618–623, 2022.

[13] Wihem Arsac, Giampaolo Bella, Xavier Chantry, and Luca Compagna. Validating security protocols under the general attacker. In Pierpaolo Degano and Luca Viganò, editors, *Foundations and Applications of Security Analysis*, pages 34–51, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[14] Sam Arts, Jianan Hou, and Juan Carlos Gomez. Natural language processing to identify the creation and impact of new technologies in patent text: Code, data, and new measures. *Research Policy*, 50(2):104144, 2021.

[15] Domagoj Babić, Stefan Bucur, Yaohui Chen, Franjo Ivančić, Tim King, Markus Kusano, Caroline Lemieux, László Szekeres, and Wei Wang. Fudge: fuzz driver generation at scale. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 975–985. Association for Computing Machinery, 2019.

[16] Cesare Bartolini, Andra Giurgiu, Gabriele Lenzini, and Livio Robaldo. Towards legal compliance by correlating standards and laws with a semi-automated methodology. In *BNCAI*, pages 1–16, 2016.

[17] Giampaolo Bella, Gianpietro Castiglione, and Daniele Francesco Santamaria. An automated method for the ontological representation of security directives. In *Proceedings of the Joint Ontology Workshops 2023 Episode IX: The Quebec Summer of Ontology co-located*

with the 13th International Conference on Formal Ontology in Information Systems (FOIS 2023), Sherbrooke, Québec, Canada, July 19-20, 2023, volume 3637 of *CEUR Workshop Proceedings*, pages 1–17. CEUR-WS.org, 2023.

[18] Giampaolo Bella, Gianpietro Castiglione, and Daniele Francesco Santamaria. An ontological approach to compliance verification of the NIS 2 directive. In *Proceedings of the Joint Ontology Workshops 2023 Episode IX: The Quebec Summer of Ontology co-located with the 13th International Conference on Formal Ontology in Information Systems (FOIS 2023), Sherbrooke, Québec, Canada, July 19-20, 2023*, volume 3637 of *CEUR Workshop Proceedings*, pages 1–12. CEUR-WS.org, 2023.

[19] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022, mar 2003.

[20] Piero A. Bonatti, Sabrina Kirrane, Iliana M. Petrova, and Luigi Sauro. Machine understandable policies and gdpr compliance checking. *KI - Künstliche Intelligenz*, 34(3):303–315, Sep 2020.

[21] Travis D. Breaux, Hanan Hibshi, and Ashwini Rao. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *Requir. Eng.*, 19(3):281–307, sep 2014.

[22] Gianpietro Castiglione. Knowledge base of validated correlations between security measures (from nis 2 directive) and attack patterns (from capec). https://github.com/gianpietroc/NIS2_Attack Patterns, 2025.

[23] Gianpietro Castiglione. MOSKAD Software for attack structuring in killchain for targeting NIS 2 Directive to offensive activities. https://github.com/gianpietroc/NIS2_Killchain, 2025.

[24] Gianpietro Castiglione and Giampaolo Bella. Compliance-Driven CWE Assessment by Semantic Similarity. In *Computer Security. ESORICS 2024 International Workshops*, pages 395–415. Springer Nature Switzerland, 2025.

[25] Gianpietro Castiglione and Giampaolo Bella. Modelling Offensive Security Killchains from Compliance Gaps with Security Directives. In *Computer Security. ESORICS 2025 International Workshops. In press.*, 2025.

[26] Gianpietro Castiglione, Giampaolo Bella, and Daniele Francesco Santamaria. Towards grammatical tagging for the legal language of cybersecurity. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ARES '23, New York, NY, USA, 2023. Association for Computing Machinery.

[27] Gianpietro Castiglione, Giampaolo Bella, and Daniele Francesco Santamaria. SecOnto: Ontological Representation of Security Directives. *Computers & Security*, 148:104150, 2025.

[28] Gianpietro Castiglione, Marcello Maugeri, and Giampaolo Bella. Poster: Machine learning for vulnerability detection as target oracle in automated fuzz driver generation. In Manuel Egele, Veelasha Moonsamy, Daniel Gruss, and Michele Carminati, editors, *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 140–145, Cham, 2025. Springer Nature Switzerland.

[29] Gianpietro Castiglione, Daniele Francesco Santamaria, Giampaolo Bella, Laura Brisindi, and Gaetano Puccia. Guiding cybersecurity compliance: An ontology for the NIS 2 directive. *Computers & Security*, 157:104617, 2025.

[30] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics.

[31] Yizheng Chen, Zhoujie Ding, Lamya Alowain, Xinyun Chen, and David Wagner. Diversevul: A new vulnerable source code dataset for deep learning based vulnerability detection. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '23, page 654–668, New York, NY, USA, 2023. Association for Computing Machinery.

[32] Danny C. Cheng and Nathalie Rose Lim-Cheng. An ontology based framework to support multi-standard compliance for an enterprise. In *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*, pages 1–6, 2017.

[33] European Commission. Types of legislation. https://european-union.europa.eu/institutions-law-budget/law/types-legislation_en, 2023.

[34] Christophe Debruyne, Harshvardhan J. Pandit, Dave Lewis, and Declan O'Sullivan. "just-in-time" generation of datasets by considering structured representations of given consent for gdpr compliance. *Knowledge and Information Systems*, 62(9):3615–3640, Sep 2020.

[35] Mina Deng, Kim Wuyts, Riccardo Scandariato, Bart Preneel, and Wouter Joosen. A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. *Requir. Eng.*, 16(1):3–32, March 2011.

[36] Isabella Distinto. Checking compliance in european tender documents through ontologies and rules. In *International Web Rule Symposium*, pages 1–9, 2012.

[37] Isabella Distinto, Mathieu d'Aquin, and Enrico Motta. Loted2: An ontology of european public procurement notices. *Semantic Web*, 7:267–293, 2016.

[38] George Drivas, Argyro Chatzopoulou, Leandros Maglaras, Costas Lambrinoudakis, Allan Cook, and Helge Janicke. A nis directive compliant cybersecurity maturity assessment framework. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1641–1646, 2020.

[39] Prokopios Drogkaris, Stefanos Gritzalis, Christos Kalloniatis, and Costas Lambrinoudakis. A hierarchical multitier approach for privacy policies in e-government environments. *Future Internet*, 7:500–515, 12 2015.

[40] European Cybersecurity Organisation (ECSO). White paper. nis 2 implementation: Challenges and priorities. https://ecs-org.eu/ecso-uploads/2025/01/ECSO-White-Paper-NIS2-Implementation.pdf.

[41] Lavanya Elluri, Ankur Nagar, and Karuna Pande Joshi. An integrated knowledge graph to automate gdpr and pci dss compliance. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1266–1271, 2018.

[42] ENISA. NIS 2 Technical Implementation Guidance. https://www.enisa.europa.eu/sites/default/files/2025-06/ENISA_Technical_implementation_guidance_on_cybersecurity_risk_management_measures_version_1.0.pdf.

[43] ENISA. NIS 2 Directive source. https://eur-lex.europa.eu/eli/dir/2022/2555/oj, 2022.

[44] ENISA. 2024 Report on the State of the Cybersecurity in the Union. https://www.enisa.europa.eu/publications/2024-report-on-the-state-of-the-cybersecurity-in-the-union, 2024.

[45] Beatriz Esteves and Víctor Rodríguez-Doncel. Analysis of ontologies and policy languages to represent information flows in gdpr. *Semantic Web*, 15(3):709–743, 2024.

[46] European Commission. EUR-Lex - 02022L2555-20221227 - EN - EUR-Lex — eur-lex.europa.eu. https://eur-lex.europa.eu/eli/dir/2022/2555, 2022.

[47] Stefan Fenz. Ontology-based generation of it-security metrics. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, page 1833–1839, New York, NY, USA, 2010. Association for Computing Machinery.

[48] Stefan Fenz and Thomas Neubauer. Ontology-based information security compliance determination and control selection on the example of iso 27002. *Information & Computer Security*, 26(5):551–567, Jan 2018.

[49] Stefan Fenz and Thomas Neubauer. Ontology-based information security compliance determination and control selection on the example of iso 27002. *Inf. Comput. Secur.*, 26:551–567, 2018.

[50] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. Methontology: From ontological art towards ontological engineering. In *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*, pages 1–8. American Asociation for Artificial Intelligence, March 1997. Ontology Engineering Group - OEG.

[51] Luciano Floridi. *The Fourth Revolution: How the Infosphere is Reshaping Human Reality*. Oxford University Press UK, 2014.

[52] Giorgos Flouris, Dimitris Plexousakis, and Grigoris Antoniou. A classification of ontology change. In *Semantic Web Applications and Perspectives*, 2006.

[53] GDPR Enforcement Tracker. GDPR Statistics — Fines by type of violation. https://www.enforcementtracker.com/?insights.

[54] GDPR Enforcement Tracker. GDPR Statistics — Highest individual fines (Top 10). https://www.enforcementtracker.com/?insights.

[55] Gianpietro Castiglione. Results of the application of GTCheck to the NIS 2 Directive. https://anonymous.4open.science/r/nis-tables-CBB1, 2023.

[56] Gianpietro Castiglione, Daniele Francesco Santamaria, Giampaolo Bella. NIS2Onto. https://github.com/gianpietroc/nis-ontology/blob/main/NIS2(v2).owl.

[57] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone. Modeling security requirements through ownership, permission and delegation. In *13th IEEE International Conference on Requirements Engineering (RE'05)*, pages 167–176, 2005.

[58] Google. Fog of War - How the Ukraine Conflict Transformed the Cyber Threat Landscape. https://services.google.com/fh/files/blogs/google_fog_of_war_research_report.pdf, 2023.

[59] W3C Community Group. EU Network and Information Services Directive (NIS2) — w3c.github.io.

[60] Ricardo Guimarães and Ana Ozaki. Reasoning in Knowledge Graphs. In Camille Bourgaux, Ana Ozaki, and Rafael Peñaloza, editors, *International Research School in Artificial Intelligence in Bergen (AIB 2022)*, volume 99 of *Open Access Series in Informatics (OASIcs)*, pages 2:1–2:31, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[61] M Mahmudul Hasan, George Kousiouris, Dimosthenis Anagnostopoulos, Teta Stamati, Peri Loucopoulos, and Mara Nikolaidou. CISMET. *Int. J. Semant. Web Inf. Syst.*, 17(1):1–24, January 2021.

[62] Erik Hemberg, Jonathan Kelly, Michal Shlapentokh-Rothman, Bryn Reinstadler, Katherine Xu, Nick Rutar, and Una-May O'Reilly. Linking threat tactics, techniques, and patterns with defensive weaknesses, vulnerabilities and affected platform configurations for cyber hunting, 2021.

[63] Erik Hemberg, Matthew J. Turner, Nick Rutar, and Una-May O'reilly. Enhancements to threat, vulnerability, and mitigation knowledge for cyber analytics, hunting, and simulations. *Digital Threats*, 5(1), March 2024.

[64] Sadaf Hina and P. Dhanapal Durai Dominic and. Information security policies' compliance: a perspective for higher education institutions. *Journal of Computer Information Systems*, 60(3):201–211, 2020.

[65] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[66] IBM. What is offensive security? https://www.ibm.com/think/topics/offensive-security.

[67] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey, 2018.

[68] Karuna P Joshi, Aditi Gupta, Sudip Mittal, Claudia Pearce, Anupam Joshi, and Tim Finin. Semantic approach to automating management of big data privacy policies. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 482–491, 2016.

[69] Kenta Kanakogi, Hironori Washizaki, Yoshiaki Fukazawa, Shinpei Ogata, Takao Okubo, Takehisa Kato, Hideyuki Kanuka, Atsuo Hazeyama, and Nobukazu Yoshioka. Tracing capec attack patterns from cve vulnerability information using natural language processing technique. In *Hawaii International Conference on System Sciences*, 2021.

[70] Daniel Martin Katz, Michael J Bommarito II au2, and Josh Blackman. Predicting the behavior of the supreme court of the united states: A general approach, 2014.

[71] Yevgeny Kazakov, Markus Krötzsch, and František Simančík. Unchain my EL reasoner. In Riccardo Rosati, Sebastian Rudolph, and Michael Zakharyaschev, editors, *Proceedings of the 24th International Workshop on Description Logics (DL 2011)*, volume 745 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[72] KnowBe4. The State of NIS2: A Fragmented Implementation Across the EU. https://blog.knowbe4.com/the-state-of-nis2-a-fragmented-implementation-across-the-eu#:~:text=This%20regulatory%20fragmentation%20stands%20in,as%20a%20whole%20were%20not, 2025.

[73] KPMG. NIS2 update - KPMG Netherlands. https://assets.kpmg.com/content/dam/kpmg/nl/pdf/2025/services/nis2-update.pdf.

[74] Zsolt Levente Kucsvan, Marco Caselli, Andreas Peter, and Andrea Continella. Inferring recovery steps from cyber threat intelligence reports. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 21st International Conference, DIMVA 2024, Lausanne, Switzerland, July 17–19, 2024, Proceedings*, page 330–349, Berlin, Heidelberg, 2024. Springer-Verlag.

[75] Aditya Kuppa, Lamine Aouad, and Nhien-An Le-Khac. Linking cve's to mitre att&ck techniques. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, ARES '21, New York, NY, USA, 2021. Association for Computing Machinery.

[76] Roger Kwon, Travis Ashley, Jerry Castleberry, Penny Mckenzie, and Sri Nikhil Gupta Gourisetti. Cyber threat dictionary using mitre att&ck matrix and nist cybersecurity framework mapping. In *2020 Resilience Week (RWS)*, pages 106–112, 2020.

[77] Langchain. LangChain. https://www.langchain.com/, 2025.

[78] Jieh-Sheng Lee and Jieh Hsiang. Patent claim generation by fine-tuning openai gpt-2, 2019.

[79] Yongjae Lee and Seungwon Shin. Toward semantic assessment of vulnerability severity: A text mining approach. In Alfredo Cuzzocrea, Francesco Bonchi, and Dimitrios Gunopulos, editors, *Proceedings of the CIKM 2018 Workshops co-located with 27th ACM International Conference on Information and Knowledge Management (CIKM 2018), Torino, Italy, October 22, 2018*, volume 2482 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

[80] Tong Li, Elda Paja, John Mylopoulos, Jennifer Horkoff, and Kristian Beckers. Security attack analysis using attack patterns. In *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–13, 2016.

[81] Swee Kiat Lim, Aldrian Obaja Muis, Wei Lu, and Ong Chen Hui. Malwaretextdb: A database for annotated malware articles. In *Annual Meeting of the Association for Computational Linguistics*, 2017.

[82] Marco Lippi, Przemysław Pałka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, and Paolo Torroni. CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service. *Artificial Intelligence and Law*, 27(2):117–139, feb 2019.

[83] Dongge Liu, Oliver Chang, Jonathan metzman, Martin Sablotny, and Mihai Maruseac. OSS-Fuzz-Gen: Automated Fuzz Target Generation, May 2024.

[84] Lockheed-Martin. Cyber Kill Chain®. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html.

[85] Michalis Avgerinos Loutsaris and Yannis Charalabidis. Legal informatics from the aspect of interoperability: A review of systems, tools and ontologies. In *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, ICEGOV '20, page 731–737, New York, NY, USA, 2020. Association for Computing Machinery.

[86] Pedro Henrique Luz de Araujo, Teófilo Emídio de Campos, Fabricio Ataides Braz, and Nilton Correia da Silva. VICTOR: a dataset for Brazilian legal documents classification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1449–1458, Marseille, France, May 2020. European Language Resources Association.

[87] Bernard Marr. The 10 biggest cyber security trends in 2024 everyone must be ready for now, Oct 2023.

[88] Roberta Matsunaga, Ivan Ricarte, Tania Basso, and Regina Moraes. Towards an ontology-based definition of data anonymization policy for cloud computing and big data. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 75–82, 2017.

[89] Masha Medvedeva, Michel Vols, and Martijn Wieling. Using machine learning to predict decisions of the european court of human rights. *Artificial Intelligence and Law*, 28(2):237–266, Jun 2020.

[90] Julian Heckmann Michael Poppe, Martin Lichtwark. OntoMetrics — ontometrics.informatik.uni-rostock.de, howpublished = https://ontometrics.informatik.uni-rostock.de/ontologymetrics/.

[91] MITRE. ATLAS. https://atlas.mitre.org/.

[92] MITRE. ATT&CK. https://attack.mitre.org/.

[93] MITRE. CAPEC. https://capec.mitre.org/.

[94] MITRE. D3FEND. https://d3fend.mitre.org/.

[95] Mark A. Musen. The protégé project: A look back and a look forward. *AI Matters*, 1(4):4–12, 2015.

[96] José Muñoz, Guillermo Esteban, Oscar Corcho, and Francisco Serón. Pproc, an ontology for transparency in public procurement. *Semantic Web*, 7:295–309, 03 2016.

[97] Nitin Naik, Paul Jenkins, Paul Grace, and Jingping Song. Comparing attack models for it systems: Lockheed martin's cyber kill chain, mitre att&ck framework and diamond model. In *2022 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–7, 2022.

[98] Umara Noor, Zahid Anwar, Asad Waqar Malik, Sharifullah Khan, and Shahzad Saleem. A machine learning framework for investigating data breaches based on semantic analysis of adversary's attack patterns in threat intelligence repositories. *Future Generation Computer Systems*, 95:467–487, 2019.

[99] National Institute of Standards and Technology. Csf 2.0. Technical report, U.S. Department of Commerce, Washington, D.C., 2025.

[100] Alessandro Oltramari, Dhivya Piraviperumal, Florian Schaub, Shomir Wilson, Norman Sadeh, and Joel R. Reidenberg. Privonto: A semantic framework for the analysis of privacy policies. *Semantic Web*, 9:185–203, 2018.

[101] Packtpub. Packtpub object extraction. https://subscription.packtpub.com/book/data/9781838987312/2/ch02lvl1sec16/extracting-subjects-and-objects-of-the-sentence.

[102] Monica Palmirani, Michele Martoni, Arianna Rossi, Cesare Bartolini, and Livio Robaldo. Legal ontology for modelling gdpr concepts and norms. In *International Conference on Legal Knowledge and Information Systems*, pages 1–10, 2018.

[103] Monica Palmirani, Michele Martoni, Arianna Rossi, Cesare Bartolini, and Livio Robaldo. Pronto: Privacy ontology for legal reasoning. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 1–18, 2018.

[104] Harshvardhan J. Pandit, Declan O'Sullivan, and Dave Lewis. Queryable provenance metadata for gdpr compliance. *Procedia Computer Science*, 137:262–268, 2018. Proceedings of the 14th International Conference on Semantic Systems 10th – 13th of September 2018 Vienna, Austria.

[105] Youngja Park and Taesung Lee. Full-stack information extraction system for cybersecurity intelligence. In Yunyao Li and Angeliki Lazaridou, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 531–539, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics.

[106] Paul Pols. Unified Killchain. https://www.unifiedkillchain.com/assets/The-Unified-Kill-Chain.pdf.

[107] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[108] Robert Pell, Sotiris Moschoyiannis, Emmanouil Panaousis, and Ryan Heartfield. Towards dynamic threat modelling in 5g core networks based on MITRE att&ck. *CoRR*, abs/2108.11206, 2021.

[109] Marcin Pietranik and Adrianna Kozierkiewicz. Methods of managing the evolution of ontologies and their alignments. *Applied Intelligence*, 53(17):20382–20401, April 2023.

[110] Ana Maria Pirca and Harjinder Singh Lallie. An empirical evaluation of the effectiveness of attack graphs and mitre att&ck matrices in aiding cyber attack perception amongst decision-makers. *Computers & Security*, 130:103254, 2023.

[111] Alun Preece. Asking 'why' in ai: Explainability of intelligent systems – perspectives and challenges. *Intelligent Systems in Accounting, Finance and Management*, 25(2):63–72, 2018.

[112] Joe Raad and Christophe Cruz. A survey on ontology evaluation methods. In *International Conference on Knowledge Engineering and Ontology Development*, pages 1–9, 2015.

[113] Hanene Boussi Rahmouni, Tony Solomonides, Marco Casassa Mont, and Simon Shiu. Privacy compliance and enforcement on european healthgrids: an approach through ontology. *Philos. Trans. A Math. Phys. Eng. Sci.*, 368(1926):4057–4072, September 2010.

[114] Niklas Risse and Marcel Böhme. Uncovering the limits of machine learning for automatic vulnerability detection. In *Proceedings of the 33rd USENIX Conference on Security Symposium*, SEC '24, USA, 2024. USENIX Association.

[115] Marcelo Rodríguez, Gustavo Betarte, and Daniel Galegari. Discovering attacker profiles using process mining and the mitre att&ck taxonomy. In *Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, LADC '23, page 146–155, New York, NY, USA, 2023. Association for Computing Machinery.

[116] Krissada Rongrat and Twittie Senivongse. Assessing risk of security non-compliance of banking security requirements based on attack patterns. *International Journal of Networked and Distributed Computing*, 6:1–10, 01 2018.

[117] Martin Rosso, Michele Campobasso, Ganduulga Gankhuyag, and Luca Allodi. Saibersoc: Synthetic attack injection to benchmark and evaluate the performance of security operation centers. In *Proceedings of the 36th Annual Computer Security Applications Conference*, ACSAC '20, page 141–153, New York, NY, USA, 2020. Association for Computing Machinery.

[118] Shanto Roy, Emmanouil Panaousis, Cameron Noakes, Aron Laszka, Sakshyam Panda, and George Loukas. Sok: The mitre att&ck framework in research and practice, 2023.

[119] Iqbal H. Sarker, Md. Hasan Furhad, and Raza Nowrozy. Ai-driven cybersecurity: An overview, security intelligence modeling and research directions. *SN Computer Science*, 2, 2021.

[120] Rainer Schmidt, Christian Bartsch, and Roy Oberhauser. Ontology-based representation of compliance requirements for service processes. In *SBPM*, volume 251, pages 1–12, 01 2007.

[121] Amit Sheth, Kaushik Roy, and Manas Gaur. Neurosymbolic Artificial Intelligence (Why, What, and How) . *IEEE Intelligent Systems*, 38(03):56–62, May 2023.

[122] Stefano Simonetto, Thijs Sebastiaan van Ede, Peter Bosch, Willem Jonker, and Ronan Oostveen. Text2weak: mapping cves to cwes using description embeddings analysis. In *The 4th Workshop on Artificial Intelligence-Enabled Cybersecurity Analytics*, 2024. 4th Workshop on Artificial Intelligence-Enabled Cybersecurity Analytics ; Conference date: 26-08-2024 Through 26-08-2024.

[123] Mikko Siponen, Seppo Pahnila, and M. Adam Mahmood. Compliance with information security policies: An empirical investigation. *Computer*, 43(2):64–71, 2010.

[124] Michal Sir, Zdenek Bradac, and Petr Fiedler. Ontology versus database. *IFAC-PapersOnLine*, 48(4):220–225, 2015. 13th IFAC and IEEE Conference on Programmable Devices and Embedded Systems.

[125] Arian Soltani, DJeff Kanda Nkashama, Jordan Felicien Masakuna, Marc Frappier, Pierre-Martin Tardif, and Froduald Kabanza. Assessing language models for semantic textual similarity in cybersecurity.

[126] Stanford Center for Biomedical Informatics Research. Protégé. https://protege.stanford.edu/, 2019.

[127] Jeremy Straub. Modeling attack, defense and threat trees and the cyber kill chain, att&ck and stride frameworks as blackboard architecture networks. In *2020 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 148–153, 2020.

[128] Octavia-Maria Sulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P. Dinu, and Josef van Genabith. Exploring the use of text classification in the legal domain, 2017.

[129] Romilla Syed. Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system. *Inf. Manag.*, 57:103334, 2020.

[130] Zareen Syed, Ankur Padia, Timothy W. Finin, M. Lisa Mathews, and Anupam Joshi. Uco: A unified cybersecurity ontology. In *AAAI Workshop: Artificial Intelligence for Cyber Security*, 2016.

[131] Lionel Tailhardat, Yoan Chabot, and Raphael Troncy. Noria-o: An ontology for anomaly detection and incident management in ict systems, 2024.

[132] Gruppo TIM. Cyber Security Report - DDoS and Ransomware threat analysis. https://www.gruppotim.it/content/dam/gt/centro-studi-tim/cybersecurity-report-2025/Cyber%20Security%20Report_ENG.pdf, 2025.

[133] Takuma Tsuchida, Rikuho Miyata, Hironori Washizaki, Nobukazu Yoshioka, and Yoshiaki Fukazawa. Automatic detection of abstract–concrete relationships between attack patterns of att&ck and capec with fine-tuned bert. In *2023 10th International Conference on Dependable Systems and Their Applications (DSA)*, pages 589–590, 2023.

[134] Fahad ul Hassan and Tuyen Le. Automated requirements identification from construction contract documents using natural language processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 12(2):04520009, 2020.

[135] European Union. Directive (eu) 2016/1148 of the european parliament and of the council. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016L1148, 2016.

[136] Mounika Vanamala, Jairen Gilmore, Xiaohong Yuan, and Kaushik Roy. Recommending attack patterns for software requirements document. *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1813–1818, 2020.

[137] Ju An Wang and Minzhe Guo. Ovm: an ontology for vulnerability management. In *Cyber Security and Information Intelligence Research Workshop*, 2009.

[138] Felix Weissberg, Jonas Moler, Tom Ganz, Erik Imgrund, Lukas Pirch, Lukas Seidel, Moritz Schloegel, Thorsten Eisenhofer, and Konrad Rieck. Sok: Where to fuzz? assessing target selection methods in directed fuzzing. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, ASIA CCS '24, page 1539–1553, New York, NY, USA, 2024. Association for Computing Machinery.

[139] R.S.I. Wilson, J.S. Goonetillake, W.A. Indika, and Athula Ginige. A conceptual model for ontology quality assessment: A systematic review. *Semantic Web*, 14(6):1051–1097, 2023.

[140] World Economic Forum. Global Cybersecurity Outlook 2025. https://www.weforum.org/publications/global-cybersecurity-outlook-2025/, 2025.

[141] Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3086–3095, Online, July 2020. Association for Computational Linguistics.

[142] F. Yip, A.K.Y. Wong, N. Parameswaran, and P. Ray. Rules and ontology in compliance management. In *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, pages 435–435, 2007.

[143] Lu Yu, Yuliang Lu, Yi Shen, Yuwei Li, and Zulie Pan. Vulnerability-oriented directed fuzzing for binary programs. *Scientific Reports*, 12, 2022.

[144] Marco Zambianco, Claudio Facchinetti, and Domenico Siracusa. A proactive decoy selection scheme for cyber deception using mitre att&ck. *Comput. Secur.*, 148(C), January 2025.

[145] Botao Zhong, Chen Gan, Hanbin Luo, and Xuejiao Xing. Ontology-based framework for building environmental monitoring and compliance checking under bim environment. *Building and Environment*, 141, 05 2018.

[146] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. Jec-qa: A legal-domain question answering dataset, 2019.

[147] Xiaogang Zhu, Shigang Liu, Xian Li, Sheng Wen, Jun Zhang, Seyit Ahmet Çamtepe, and Yang Xiang. Defuzz: Deep learning guided directed fuzzing. *CoRR*, abs/2010.12149, 2020.

Acknowledgments

Terminare il percorso del dottorato equivale a terminare un capitolo nella propria vita che inevitabilmente è irripetibile. Si forgia il proprio futuro sulla base degli anni che si trascorrono all'interno di tale così lungo percorso universitario, come nessun'altra esperienza probabilmente ne avrebbe l'ambizione. Per tale motivo, e diversamente dal passato, questa tesi di dottorato necessita di essere completata con questa sezione.

Wholeheartedly thanks al Prof. Giampaolo Bella, per essere guida scientifica e non, per il rigore accademico e la postura comportamentale, a cui ambire e che mi hanno guidato sin dalla tesi di laurea triennale. Per le esperienze che mi ha permesso di vivere in maniera assolutamente unica durante questo percorso di dottorato, al di là di ogni mia previsione e immaginazione. Un ringraziamento che non sarà mai sufficiente per la riconoscenza che vuole esprimere.

Un sincero ringraziamento all'azienda Intrapresa S.r.l. per aver cofinanziato la borsa che mi ha permesso di iniziare e portare a termine questo percorso. All'Ing. Gaetano Puccia per la pronta ospitalità, la gentilezza e la cordialità.

Grazie al gruppo NaS.Inf per l'amicizia e il lavoro di squadra. Grazie a Daniele per la preziosa collaborazione: la sua esperienza e competenza nel campo delle ontologie sono state fondamentali per affinare e migliorare questo lavoro. Un ringraziamento speciale a Mario e Marcello: gli alti e bassi che questo percorso inevitabilmente comporta, le prospettive future e le analisi sul passato e presente sono, senza dubbio, quanto ci ha accomunato di più.

Un profondo ringraziamento al Prof. Andrea Continella e al gruppo SCS dell'Università di Twente per la meravigliosa esperienza vissuta, accademica e non, e da cui sono nati inaspettati legami.

Infine, voglio concludere *il mio gemello di carta e inchiostro* con una sentita riconoscenza alla vita. Alle proprie esperienze e relazioni, agli arrivi e alle perdite, alla famiglia e alle amicizie di sempre. Ai desideri, alle aspirazioni, ai tormenti, alle occasioni e agli errori, alle passioni e vulnerabilità. All'eleganza e alla nobiltà d'animo.

Alle tenebre e alla luce.

Al passato, al presente, e al futuro. Alla somma di tutte le caratteristiche che nella propria razionalità e irrazionalità prendono vita, si sviluppano e si intrecciano, definitivamente rendendo singolarità.

E a te, se sei rimasto fin proprio alla fine.

– J.K. Rowling

La vita quando passa sembra un
fiume dove mi muovo
Guardo un'altra volta il cielo per
sapere se c'è
Una stella che mi guidi ancora
fuori da quel che sono
Fuori da me
Fuori da me.

— XVI Religion

And I would like to be able to
continue to let what is inside of me
Which is, which comes from all the
music that I hear
You know, I'd like for that to come
out
And it's like, it's not really me
that's coming
The music's coming through me
The music's coming through me.

— DJ Shadow