XAI IN CYBERSECURITY

TOWARDS EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI) IN CYBERSECURITY

By EDUARDO LOPEZ, B.Eng., MBA

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements for the Degree Doctor of philosophy

McMaster University
DOCTOR OF PHILOSOPHY (2024)
Hamilton, Ontario, Canada (DeGroote School of Business)

TITLE:	Towards eXplainable Artificial Intelligence (XAI) in Cybersecurity
AUTHOR:	Eduardo Lopez B.Eng. (Electronics), Universidad Javeriana, Colombia MBA, McGill University, Canada
SUPERVISOR:	Dr. Norm Archer Dr. Kamran Sartipi Dr. Yufei Yuan
NUMBER OF PAGES:	xvii, 161

Lay Abstract

Artificial Intelligence (AI) is now pervasive in our lives, intertwined with myriad other technology elements in the fabric of society and organizations. Instant translations, complex fraud detection and AI assistants are not the fodder of science fiction any longer. However, realizing its benefits in an organization can be challenging. Current AI implementations are different from traditional information systems development. AI models need to be trained with large amounts of data, iteratively focusing on outcomes rather than business requirements. AI projects may require an atypical set of skills and significant financial resources, while creating risks such as bias, security, interpretability, and privacy.

The research explores a real-life case study in a mid-size organization using Generative AI to improve its cybersecurity posture. A model for successful AI implementations is proposed, including the non-technical elements that practitioners should consider when pursuing AI in their organizations.

Abstract

A 2023 cybersecurity research study highlighted the risk of increased technology investment not being matched by a proportional investment in cybersecurity, exposing organizations to greater cyber identity compromise vulnerabilities and risk. The result is that a survey of security professionals found that 240% expected growth in digital identities, 68% were concerned about insider threats from employee layoffs and churn, 99% expect identity compromise due to financial cutbacks, geopolitical factors, cloud adoption and hybrid work, while 74% were concerned about confidential data loss through employees, ex-employees and third party vendors. In the light of continuing growth of this type of criminal activity, those responsible for keeping such risks under control have no alternative than to use continually more defensive measures to prevent them from happening and causing unnecessary businesses losses. This research project explores a real-life case study: an Artificial Intelligence (AI) information systems solution implemented in a mid-size organization facing significant cybersecurity threats. A holistic approach was taken, where AI was complemented with key non-technical elements such as organizational structures, business processes, standard operating documentation and training - oriented towards driving behaviours conducive to a strong cybersecurity posture for the organization. Using Design Science Research (DSR) guidelines, the process for conceptualizing, designing, planning

and implementing the AI project was richly described from both a technical and information systems perspective. In alignment with DSR, key artifacts are documented in this research, such as a model for AI implementation that can create significant value for practitioners.The research results illustrate how an iterative, data-driven approach to development and operations is essential, with explainability and interpretability taking centre stage in driving adoption and trust. This case study highlighted how critical communication, training and cost-containment strategies can be to the success of an AI project in a mid-size organization.

This is for Moni, Andrea and Nico

And also for Me

Acknowledgements

This research, as the culmination of the Ph.D. program, would not have been possible without the patience, support and encouragement of my family. The decision to enter the program, and the resiliency and persistence required to complete it can be directly traced to the uncompromising love of Monica, my wife, every step of the way - always an unrelenting source of music and colour. From my daughter Andrea I learned the courage to make the right decisions, notwithstanding the difficulties, and always true to core values. From my son Nicolas I learned the discipline to pursue my goals with determination and sacrifice - with a righteous path every bit as important as a worthy destination. From my mother, Omaira, that was there when I needed her during those dark days, and that instilled in me the unbreakable belief in the value of intellectual pursuit.

I am indebted to the members of my supervisory committee. Their knowledge, patience and support for an atypical student during the preceding 6 years cannot be understated. They saw in me a student worth teaching to, and believed in me even when I lacked that belief myself. Thank-you for making me a better professional, an academic and for the wise guidance I received along the way.

Maarif, Bell, Mehmet - your friendship made the Ph.D. a joyous memory to hold forever.

Table of Contents

List of Figures

List of Tables

Notation, Definitions, and Abbreviations

Notation

$A \leq B$ A is less than or equal to B

Definitions

Generative AI

Generative AI refers to artificial intelligence systems that can create content such as images, audio, text or videos. In contrast, most of the AI implementation to date could be categorized as discriminative AI which is focused on classification and prediction.

JSON It is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays. It is a common data format with diverse uses in electronic data interchange, including that of

web applications with servers.

Phishing It is a malicious activity where individuals are contacted by email, telephone or text message by someone posing as a legitimate party, and with the intention of obtaining sensitive data such as personally identifiable information, financial details details, and passwords.

Prompt engineering

Prompt engineering is the process of structuring text that can be interpreted and understood by a generative AI model. A prompt is natural language text describing the task that an AI should perform.

REST API Also known as RESTful API, is an application programming interface that conforms to the constraints of REST architecture. REST is a software architectural style that was created to guide the design and development of the architecture for the World Wide Web. REST defines a set of constraints for how the architecture of a distributed, Internet-scale hypermedia system, such as the Web, should behave.

Abbreviations

AI Artificial intelligence

CIA The CIA triad refers to confidentiality, integrity and availability, describing a model designed to guide policies for information security (infosec) within an organization. The model is sometimes referred to

as the AIC triad – which stands for availability, integrity and confidentiality – to avoid confusion with the Central Intelligence Agency.

GxP It is a general abbreviation for the "good practice" quality guidelines and regulations. The "x" stands for the various fields, including the pharmaceutical and food industries, for example good agricultural practice, or GAP. A "c" or "C" is sometimes added to the front of the initialism.

LLM Large Language Models are Artificial Intelligence (AI) models that have been trained on Internet-scale datasets and produce natural language responses.

RAG It is an AI framework for retrieving facts from an external knowledge base to ground large language models (LLMs) on the most accurate, up-to-date information and to give users insight into LLMs' generative process.

SSO Single sign-on is an authentication scheme that allows a user to log in with a single ID to any of several related, yet independent, software systems. True single sign-on allows the user to log in once and access services without re-entering authentication factors.

TOGAF The Open Group Architecture Framework is the most used framework for enterprise architecture as of 2020 that provides an approach for designing, planning, implementing, and governing an enterprise information technology architecture. TOGAF is a high-level approach to design.

UI In the industrial design field of human–computer interaction, a user interface is the space where interactions between humans and machines occur.

Chapter 1

Introduction

1.1 Background

AI encompasses a large number of different technologies, techniques and approaches to solving analysis challenges. First conceptualized in the 1950s [2], progress in the field has not been linear. Significant periods of hype were followed by long stretches of time where disappointment pervaded academia and industry – known as the "AI winters" by the cognoscenti. The current instantiations of AI use data as the essential raw material for delivering results [62]. This was not always the case: AI was initially based on expert systems with interpretable rules. However, AI has now gravitated towards machine learning techniques - creating algorithms that learn from examples before analyzing new data.

The last winter ended - arguably - in early the 2010s, with the advent of a particular technology: Deep Learning. Based on neural networks, deep learning uses multiple hidden layers of neurons that are trained in relevant datasets [5]. The operations performed by each neuron are mathematically simple and easily understood,

but when analyzed as a whole, the number of parameters quickly grow beyond the ability of humans to understand it. Although not publicly known, several sources put the number of parameters in ChatGPT to be around 1.7 trillion[70]. This level of complexity makes AI effectively a black-box, potentially limiting the usability of the system in some organizational processes.

An additional dynamic makes AI a remarkable research domain. Designing and implementing information systems has typically followed a structured, linear approach. In some regulated industries such as pharmaceuticals and finance, software programmers depart from a set of documented business requirements and declaratively write code to fulfill them. It is for the most part a transparent and predictable process. In contrast, AI projects mix linear and non-linear approaches where data acquisition, creation and/or manipulation play an essential role, followed by somewhat unpredictable trial and error iterative experiments where variables (called hyperparameters) are adjusted until results are deemed satisfactory[89].

The risks for society and organizations in the use of AI are significant. An example offers a good illustration of the challenges: IBM's Watson. The company's approach was quite effective initially - starting with "Deep Blue" in 1997 [47], IBM developed systems that could beat human contenders in games where human intelligence was the fundamental raw material. Using the name "Watson" to group multiple AI technologies, IBM designed and implemented a system to compete in the popular show *Jeopardy!*. Watson was very successful at achieving this objective. It was able to beat the best human player known to date: Ken Jennings [45]. Succeeding in Jeopardy! required significant ingenuity in combining multiple advanced AI concepts in Natural Language Processing (NLP), information retrieval and automated reasoning, to name

a few. Its success heralded a new era where AI could be applied to domains where data was readily available, so IBM set its sights in a field broadly recognized as a gigantic business opportunity: healthcare. IBM proceeded to build Watson health into a significant business concern - using the technology piloted through Jeopardy! and acquiring massive amounts of data (and the companies that owned them, for approximately USD$5 billion), IBM built a business unit of 7,000 employees to revolutionize healthcare [68].

Fast-forward to 2022 and a very different result was achieved. IBM sold Watson health in parts to multiple companies, getting only US$1 billion and conceding defeat in one of the most dramatic flops for AI since the last AI winter [67]. As it is the case usually, there was not a single reason for this outcome. There were many errors, strategic, tactical and operational that IBM made during the preceding 10 years. However, some of the most interesting ones can be linked to factors described in the preceding section. There was a deluge of data, but it was not structured or homogeneous enough to use in training the AI model. In fact, insights that may apply to patients in one geographic location or people cohort (e.g., age or race) cannot be just transferred to another one. The model that was created through iterative experimentation for Jeopardy! did not work for a different domain where complexity was abundant and data quality and availability a fundamental challenge. IBM Watson's failure will be studied for a long time by industry and academia. But it highlights one of the key research problems that this research explores: implementing trust-worthy AI systems.

1.1.1 Theoretical sensitivity

The research presented in this document is located at the intersection of very different domains. This diversity may pose some challenges, but also a rich research opportunity for contribution to practitioners. I believe my experience, skills and knowledge can be of great use to the exploration of the dynamics.

Philosophical world views fundamentally guide how researched is performed. A positivist paradigm assumes that there are relationships that can be proven or falsified and that exist independent of the observer. In contrast, an interpretive perspective understands reality as a human construction, and is therefore influenced by the researcher when analyzing phenomena [64]. Constructivist research approaches in social sciences – such as Grounded Theory – explicitly considers the researcher's theoretical sensitivity as an essential component of the value provided [90, 103].

I am an Information Technology (IT) professional, with more than two decades of technical, managerial and executive experience enabling business processes with the cost-effective use of the right information technologies. I have been accountable for IT departments in multiple organizations across several industries (some of them regulated such as pharmaceuticals). An ever growing part of my roles have included cybersecurity. Complementing my experience, I acquired significant expertise in data science and machine learning. I believe that my profile, as a researcher, can contribute to the current discourse, and advance knowledge in very important areas.

1.2 Theory in information systems

Theory plays a fundamental role in research. The research domains influence the nature of theory - both when it is used or produced by a study. In order to establish the framework under which my research is performed, it is important to understand the typology of theories in information systems. The seminal article published by Shirley Gregor in MIS Quarterly establishes the following five categories [34]:

- Theory for analysis, where no causal relationships are explored, nor predictions offered.
- Theory for explanation, that shows the relationships without the objective of predicting or testing propositions.
- Theory for prediction, focusing mostly on what the results are expected to be, without much emphasis in the explanation of the phenomena.
- Theory for explanation and prediction, with causal explanations, testable propositions and predictions.
- Theory for design and action, with prescriptions on the construction of artifacts that seek to achieve a goal.

The complex, multifaceted phenomena that is explored in this work departs from existing theories, many of them in natural sciences and engineering. The objective is not to prove or test existing theories, but rather explore the different dynamics in AI creation under a defined domain, and distil potential best practices that can support practitioners in the future. As will be explained in Chapter 3, I focus on establishing prescriptions for construction of artifacts that effectively achieve the goals.

The models, methods and instantiations created as part of this work use as departing point both natural and social theories which are articulated in the literature review in chapter 4.

1.3 Structure

The structure of this document emphasizes its design science orientation while ensuring the foundational literature is thoroughly reviewed and is relevant for both engineering and social scientists, in practice or academia.

After the preface and this introductory chapter, I proceed to explain the research problem, questions and scope in chapter 2, followed by the methodological aspects in chapter 3. By bringing forward these two topics before the extensive literature review, I intend to provide the reader with holistic context to evaluate the research.

Chapter 4 provides the theoretical background that supports this research. The chapter introduces the knowledge domains, followed by the relevant elements on the inherently complex phenomena of interest. This chapter includes a perspective on AI evolution towards the current state of the art, later providing complementary perspectives on eXplainable Artificial Intelligence (XAI), cybersecurity and Information Technology (IT) governance.

Chapter 5 provides the results of the research. In alignment with the research methodology, the resulting artifacts and instantiations are described. An overarching model of eXplainable AI (XAI) implementation is posited that is based on the experiences gained through the process of implementing AI for cybersecurity in an organization, and that can be used as the departure point of a theory for design and action. Chapter 6 concludes this document.

Chapter 2

Research questions and scope

2.1 Research problem

A very large amount of resources is spent every year on information security. Despite so much attention, cybersecurity issues continue unabated, with malicious actors finding ever more sophisticated ways to inflict damage. The attack surface is also an increasing parameter. Higher degrees of connectivity and technology penetration create unknown risks in virtually every human endeavor involving information security.

Organizations respond to this environment by designing, implementing and monitoring key controls. Medium- and large-size organizations usually have fully dedicated resources - both human and technological - to contain the threats and mitigate impact. With the advent of Artificial Intelligence (AI), a new and powerful toolbox is now in place, enabling organizations to address more effectively and efficiently cybersecurity threats.

However, using AI presents multiple challenges. As with any new technological advance, there is a new skill set required that is not readily available. The concepts

historically applied to declarative programming do not extend transparently to environments where data effectively drives the programming, such as it is the case with machine learning in AI. Furthermore, the superior results achieved by AI may be obscured by the low interpretability, failing to create trust in the user. In a domain space such as cybersecurity, failure to create trust in the results of the system would dramatically impact adoption. This is the problem that this research explores.

2.2 Research purpose and questions

The purpose of this research is to explore how AI can be designed and implemented by a cybersecurity practitioner, realizing the benefits from the use of recent advances in machine learning, while enabling interpretability, effectiveness and efficiency of the practitioner's tasks, ultimately generating trust in the system.

In alignment with the research proposal previously documented, but taking into consideration the immense speed of new advances in the topic, the following are the research questions explored.

> **RQ1:** How can state of the art Artificial Intelligence be used for characterizing and identifying potential cybersecurity threats taking place in an organization?

It is important to highlight that this research question is multi-dimensional, involving complex domains and their interactions, and requires a line of inquiry that is adjusted and enhanced continuously based on the intermediate outputs of the process.

> **RQ2:** What strategies are effective in the design and implementation of an AI system, so it optimizes interpretability and generates trust and

adoption.

2.3 Research approach

The phenomena of interest revolves around the creation and use of AI for the purposes of cybersecurity. The intention is to elicit understanding of the underlying dynamics in the search for suitable solutions. The domain cannot be reduced to a set of variables and relationships that are quantified. In the search for eXplainable Artificial Intelligence (XAI) in cybersecurity we depart from the collection of data in a natural setting. In order to understand the situations present in the creation of AI artifacts, the researcher knowledge and experience are key instruments in the research. A holistic approach is taken in this work, traversing the continuum between natural sciences and engineering, towards elements more associated with information systems such as IT governance and design sciences.

This research is rooted on a real life case study, with an emergent design that adjusts as more information becomes available. It requires deep exploration of the technology itself, since it both constrains and enables different facets of the information system implementation.

2.4 Research contributions

Designing and implementing an AI system in the present time is more akin to art than to science. Despite the well-defined theories in engineering and mathematics that AI typically employs, the non-linear approach that must be used is not well articulated or documented. Furthermore, the low interpretability inherent in AI systems can

impact user adoption. With these challenges in mind, the following are the key research contributions offered through this research.

2.4.1 Fit-for-purpose, fit-for-use AI architecture for improving an organization's cybersecurity posture

This research richly details how a multi-source, multi-model AI architecture optimized for fast response and interpretability can be deployed in a cost-effective manner while using state-of-the art technologies such as Generative AI (GenAI) [55].

2.4.2 Critical factors in the design and implementation of AI

Through the case study, and the comparing and contrasting of the building of an AI system vis-à-vis a traditional Systems Development Life-cycle (SDLC), practitioners will gain an improved understanding of best practices to employ in a real-world setting.

2.4.3 Best practices in the building of interpretable AI systems in organizations

Using design science as the backdrop [48], this research provides the insights that practitioners may use to build AI systems that drive explainability and interpretability.

Chapter 3

Methodology

3.1 Introduction

In alignment with the inherent complexity of the phenomena, and the diversity of the domains involved, this section articulates how the methodology that is used enables rich exploration of the topics through a case study, oriented by design science principles applied to information systems research. To best understand the methodological context, it is essential to understand the fundamental role that theory plays in information systems research. Theorizing revolves around articulating explanations and predictions that can be tested [37]. The work by Dubin specifies seven components of a theory including the units whose interactions are the focus of the research and the laws of interaction between those units [24]. In the information systems domain, a theory may belong to one of five types: analysis, explanation, prediction, explanation and prediction, design and action [34]. Each of these types has specific attributes that influence the role the theory plays within the research context.

There are three research approaches used in social sciences: qualitative, quantitative and mixed methods [14]. The selection of a research approach is influenced by multiple factors, including the research problem, the intended audience for the study and the researcher's experience. Selection of an approach departs from the philosophical paradigm that drives the role of theory in the research. A *positivist* paradigm is typically deterministic and reductionist – a theory exists and the research objective is to verify it through data observation. Under this paradigm the research approach is quantitative, with methods based on instruments or survey responses statistically analyzed seeking verification of the theory.

In contrast, a qualitative approach is used when there is no theory, or where the confluence of factors is too complex to be reduced to a set of variables and relationships. In this type of research, the data are collected in their natural setting from multiple sources such as documents, interviews, observations and audiovisual information. The researcher is a key instrument, interpreting the information gathered by leveraging their personal background, culture, knowledge and experience [80]. In qualitative studies the research approach is adjusted as more information becomes available, with the objective of providing multiple perspectives reflective of the underlying complexity of the phenomena under study [15].

The information systems domain has a rich history of research with a behavioural science orientation. However, a complementary viewpoint has emerged that focuses on artifacts creation: design science research [36, 42, 48]. In a design-science orientation, the knowledge, meaning and understanding of a given context is enriched by the building of the artifacts that address the problem [1].

The phenomena I explore in this work revolves around the design and implementation of an eXplainable Artificial Intelligence (XAI) system in a real life organizational context, explored in detail through a case study addressing a critical need. It is inherently multi-dimensional and complex, defying reduction into a set of units with testable hypotheses and laws of interaction that are heralded as hallmarks of good theories [97]. Thus, the approach I take is qualitative, oriented towards design science as there are multiple artifacts - some of them technological – that are needed for the ultimate objective of improving an organization's cybersecurity posture. I take advantage of my own knowledge and experience in Information Technology (IT) governance and state of the art AI for richly discovering the multiple facets that play a role in the phenomena of interest. The fundamental paradigm I use is constructivist, where my own experiences shape meaning and understanding of the dynamics at play, effectively becoming a key instrument in the research [29]. I use a case study for the exploration of meaning, a line of inquiry that offers a remarkable opportunity to understand the multi-layered elements present in the design and evaluation of artifacts for cybersecurity [95].

3.2 Design science in information systems research

The Information Systems Research Journal states in its editorial statement and policy that the goal of information systems research is to "... expand the knowledge enabling the application of IT to human organizations and their management". Many authors have offered their interpretations on what constitutes a contribution to knowledge, including partial or incomplete theory [97] and how it is formalized. Adding to the knowledge base requires not only behavioral research rooted in the natural sciences

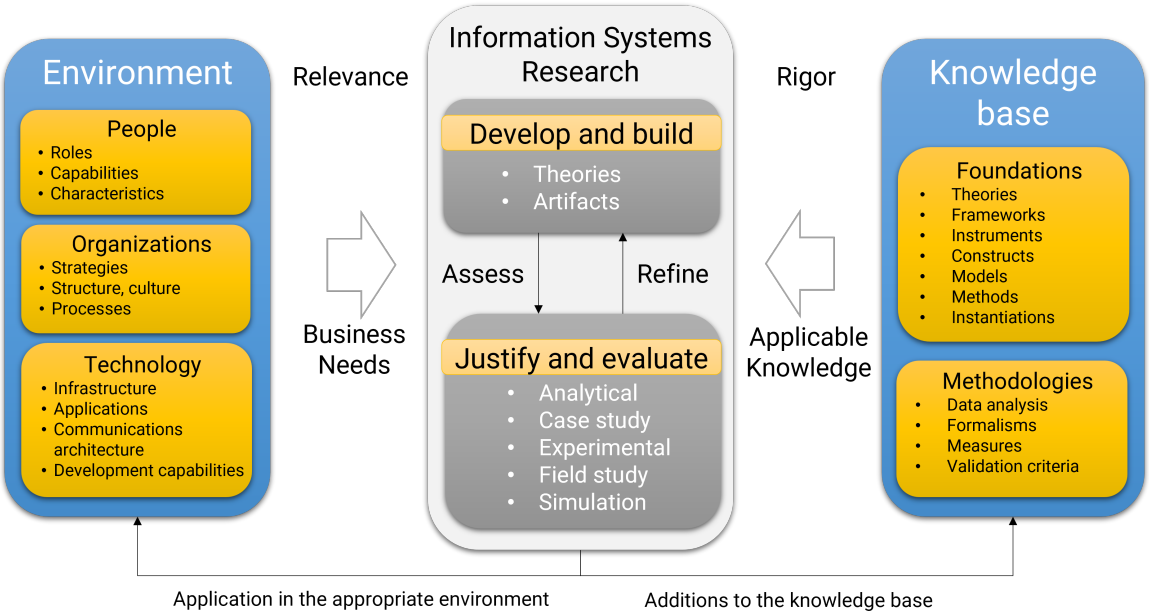


Figure 3.1: Hevner's information systems research framework, adapted from image in [42].

methods, but also research based in engineering and the sciences of the artificial [92]. Design science research contributes to the creation of knowledge with the creation of *artifacts*. Whereas a theory is more abstract and has no material existence, artifacts can be readily converted to a material existence – such as an algorithm converted to operational software. Research contributions from artifacts may include constructs, design principles or implicit technological rules [35].

To illustrate the complementary natures of behavioural and design science research, Hevner presents a rich framework [42] that is depicted in Figure 3.1.

3.2.1 Environment

The Environment pertains to the context in which the phenomena of interest exists (i.e. the problem space). It is defined based on its three components: People,

Organizations and Technologies [42].

People

This component of the environment is defined by the individuals participating in the dynamic. This includes the roles, the capabilities that are required, and the characteristics needed. Both the behavioral and design-science orientations highlight the importance of people in information systems research. Identifying cybersecurity threats (which is the objective of this exercise) cannot be accomplished solely through automated processes - it needs the active participation of individuals. These individuals shall play well-defined roles and must have, therefore, the appropriate capabilities to perform the needed tasks.

Organizations

I explore the dynamics that take place in mid-to-large organizations where the use of information systems is pervasive, and sufficient resources are allocated to cybersecurity tasks. Such an organization has well-defined strategies, with organizational units (i.e., structure) performing predefined, controlled processes. The organizational structure and its associated processes are usually represented in artifacts such as controlled documentation. This may include organizational charts, Standard Operating Procedures (SOP) and policies that govern the use of the information systems.

Technology

The final component in the environment is technology. It is a fundamental building block in this research, exploring how Artificial Intelligence (AI) is used to fulfill the

cybersecurity needs of the organization. The technology used includes the underlying infrastructure powering the applications used, as well as the communication technologies and development capabilities.

The three environmental components articulated by the design-science research framework are used for the formulation of the *business needs* as depicted in Figure 3.1.

3.2.2 Knowledge base

The right side of Figure 3.1 depicts the knowledge base for the research. The research orientation - behavioural or design-science - will drive the use of different elements of the knowledge base.

Foundations

When the research orientation is behavioural, the knowledge base includes behavioral theory components, with frameworks and instruments enabling data collection and analysis techniques. In contrast, a design science orientation gravitates towards computational and mathematical theories, frameworks and metrics. In the case of AI, the foundations and theories are at an embryonic stage. The opacity of the mechanics achieving suitable results is a remarkable opportunity for the continued discovery of new foundations that can be applied. The current state in AI is powered by the significant advances in hardware and the resulting parallel processing [5, 31, 63]. These foundational concepts are richly described in the literature review on chapter 4.

In addition to the technology foundations, multiple other elements are required, forming a dense meaningful fabric that includes artifacts from Information Technology

(IT) governance and cybersecurity. This holistic approach to meeting organizational needs formulated from the environment, seeks to increase the value of this research and broaden the potential audience that can benefit from it.

Methodologies

A design-science research orientation influences the methodologies that can be applied to the problem space. While behavioural research uses quantitative data analysis based on instruments (i.e., surveys) created by the researcher [73], design-science research leans towards evaluation of efficiency and/or effectiveness for the artifacts produced [1]. The diverse set of artifacts designed and created in the case study creates a remarkable opportunity for the exploration of potential evaluation methods that can be applied by practitioners under similar circumstances [46].

Building and evaluation

The applicable knowledge needed to meet the business needs is used for the actions taken by the researcher. In more traditional behavioural sciences research the focus is on the development of theories that can be justified. This is in contrast with design science research where the intent is mostly the building of artifacts that are rigorously evaluated [41].

The output from design science research is *design theory*: both *how* to undertake the building of an artifact and *what* the artifact looks like when built [33]. It is important to emphasize that design science research differs from routine information system built when it addresses the problem in unique or innovative ways, adding to the knowledge base, or as it is the case here, when the nature of the problem is itself

quite new, requiring new applications for existing tools and methodologies [35].

3.2.3 Thesis design-science research principles

Figure 3.2 depicts key elements that ensure a study can be considered design science research [35]. I will proceed to explore each of them in the context of the work presented in this study.

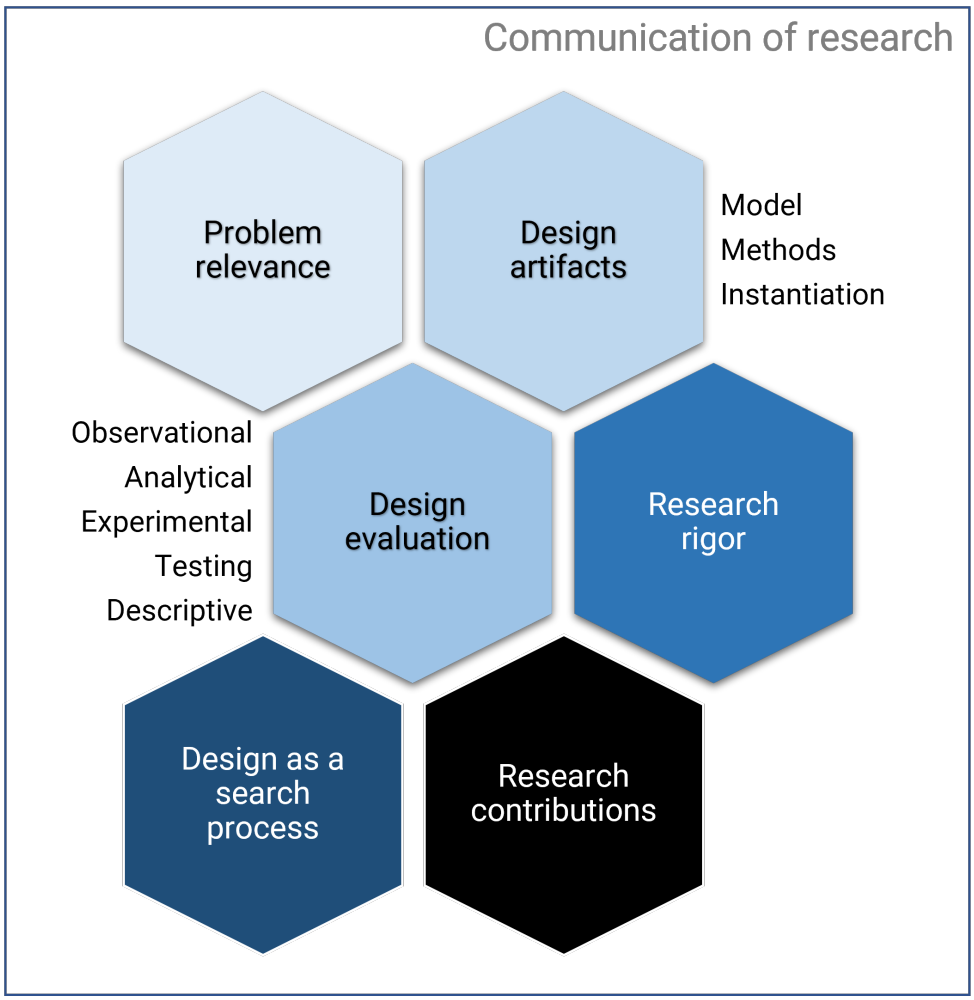


Figure 3.2: Producing design science research, created by E. Lopez, adapted from information in [1].

Problem relevance

The goal of design science research is utility – producing technological solutions to relevant problems[42]. Cybersecurity is a very current and significant challenge in highly-technified societies. The World Economic Forum published a global cybersecurity perspective that provides insights into the current state. As stated in the report, 86% of business leaders surveyed believe that a significant cybersecurity event will create a large disruption within the following 2 years [30]. At the more micro level, there is hardly a day when a cybersecurity issue does not take place - the Security magazine estimates that there are over 2,200 attacks every day. The problem is real, relevant, current and supremely important for organizations and society.

Design as an artifact

Design science research must produce artifacts in the form of constructs, models, or instantiations [42]. A **model** represents real world situations – the underlying problem and the solutions articulated – through constructs or language eliciting understanding and communication of the problems and solutions posited by the research. **Methods** are the process definitions of the problem solving performed in the research. Methods can exist in a continuum from very formal (e.g., mathematical algorithms) to informal such as textual description of best practices. An **instantiation** demonstrates how the models and methods can be used to create a working system. The research presented in this document produces artifacts in alignment with the guidelines: model, methods and instantiations. Using a case study – explained in section 4.4, a very complete and comprehensive situation will be used for the development of models and methods [1].

Design evaluation

In design science research, the evaluation of the design must be rigorously performed through suitable evaluation methods [1]. I approach the evaluation of the design through observational and analytical assessments, using real datasets found in practice. I perform experiments and rigorous testing to demonstrate suitability and effectiveness, ensuring the design creates the value needed for addressing the organization's needs.

Research contributions

Design science research must produce tangible contributions to the domain being investigated. Relevant additions to the knowledge base ensure the research further evolves the domain, creating value for practitioners [35]. The contributions of the research are described in section 2.4, with abundant details in the results section, chapter 5.

Research rigor

As in the case with behavioural research, design science research is very dependent on the suitable selection of techniques for its execution. As in the case with this research, eXplainable Artificial Intelligence (XAI) techniques are chosen based on solid theoretical foundations and relevant guidelines applied to the domains being explored. The designed artifacts in this work are an essential element of a human-computer system, designed and explained to elicit understanding on why they work so new artifacts can be created in the future for use in similar problem spaces.

Design as a search process

The work documented in this study is the result of iterative design, as is inherent in design science. XAI is intrinsically iterative – significantly more so than declarative software where rules are explicit and captured in the algorithm. Since XAI is fundamentally the creation of a learning artifact, its effectiveness and efficiency are governed by variables (i.e., hyper-parameters) that are fine-tuned based on interim results and metrics. I explore these dynamics in the results section of this document.

Communication of research

Design science considerations must be presented to audiences composed of information systems and information technology individuals. Thus, it requires an appropriate level of technical detail while maintaining it in an organizational context. The work documented in this study conveys both types of messages in a holistic manner but ensuring that an adequate exploration of the management and technical aspects exist. This approach ensures the research project can be reproduced and extends the knowledge base for the domain [41].

Chapter 4

Review of literature

4.1 Introduction

Exploring the phenomenon associated with the *design and implementation of eXplainable Artificial Intelligence (XAI) for cybersecurity in organizations* requires understanding of multiple intersecting knowledge domains, each of them significant in their own discourses and with a relatively high level of complexity. To best describe the current state for each of them, this chapter will delve into each and their most relevant elements, as depicted in figure 4.1.

Upholding the confidentiality, integrity and availability of information assets in an organization is both a complex task and a difficult objective to achieve. The literature review is structured around four major domains that are developed in detail in the current chapter. The first one provides the context from the focus domain application in this research: *cybersecurity*. The next chapter provides increased specificity by exploring how information technologies are implemented in an organization through an *Information Technology governance* framework. The final two chapters traverse

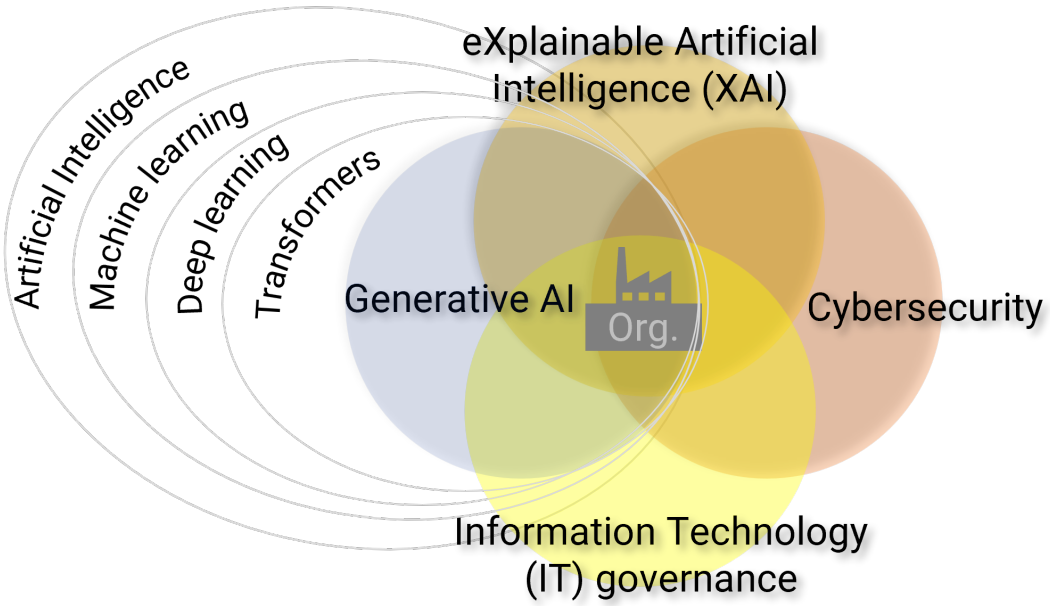


Figure 4.1: Research domains for thesis, created by E. Lopez.

towards more technical domains - starting with *Artificial Intelligence (AI)* and its recent explosion in the collective consciousness, closing with relevant elements for eliciting understanding and interpretability through *eXplainable Artificial Intelligence (XAI)*.

4.2 Cybersecurity

The ever increasing technification of society and organizations have brought an unintended byproduct: the opportunity for malicious actors to take advantage of technological progress to advance their agendas. The attack surface has dramatically increased as dependency on information technologies reaches new levels across the world.

The last five years have seen a complete change in the way mankind approaches

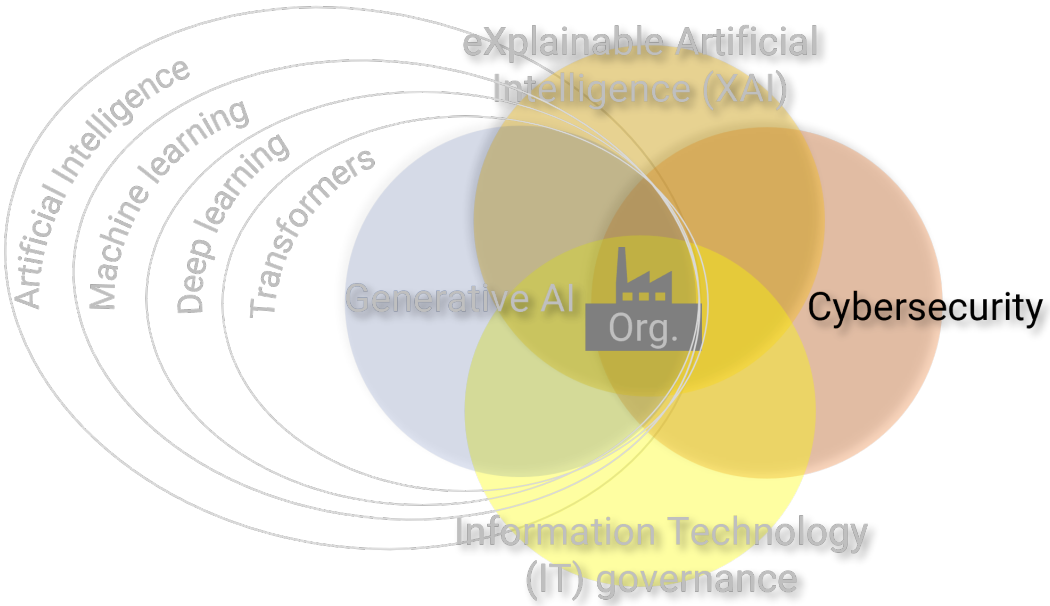


Figure 4.2: Cybersecurity in context, created by E. Lopez.

living and working. The pandemic that started at the beginning of 2020 drove levels of remote work that were deemed unrealistic just months before, further entrenching the reliance on technology for fulfilling basic needs.

The types of attacks have morphed overtime, a reflection of the changing sociopolitical dynamics and the penetration of technology in the multiple organizational layers. What started as relatively unsophisticated tactics implemented by unstructured hacking attempts is now complemented by advanced threats that move relatively slow, well organized and may be supported by nation-states. This is the case of an Advanced Persistent Threats (APT) that patiently understood, penetrated and strove to exit undetected from their targets' information systems [3]. The ATP threat actors take advantage of known exploits but also of zero-day threats, or unknown vulnerabilities for which a mitigation does not exist. One of the most recent examples revolves around the Russian invasion of Ukraine. Industry leader CrowdStrike describes in its

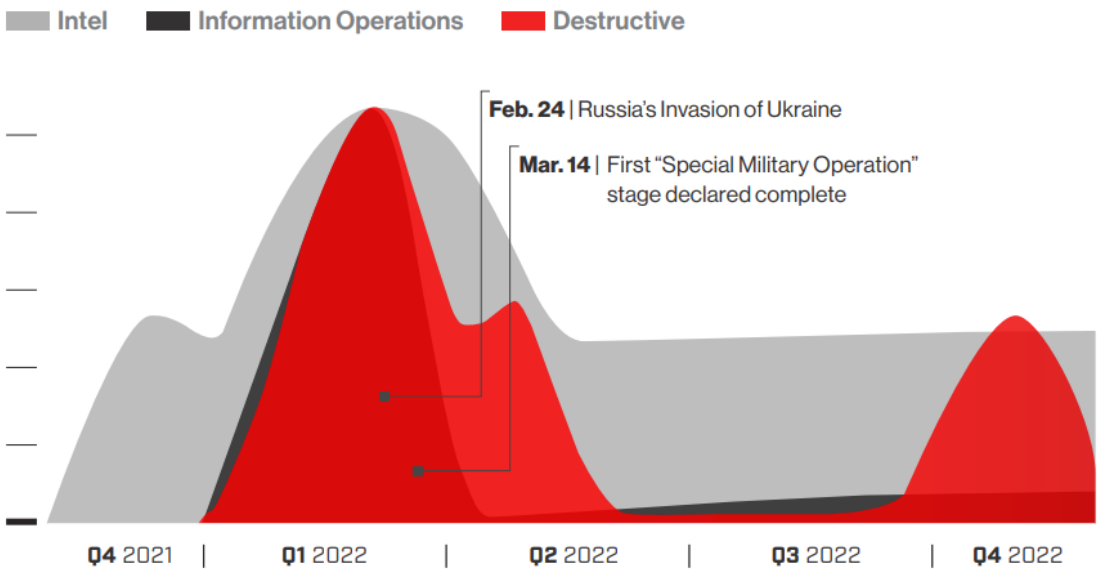


Figure 4.3: Russian operational activity against Ukraine, copied from source [16].

global threat report how Russia worked on multiple cyber-fronts to gain information and find opportunities to inflict damage.

Other sobering metrics from industry paint a challenging picture for practitioners in the organizational context. Hackers use a variety of tactics to gain entry to information systems. This may range from brute-force attacks to phishing emails that trick unsuspecting users into sharing their valid credentials. A review of the dark web shows a worrying increase in access brokers - threat actors that sell valid credentials for accessing organizational information systems. From 2021 to 2022 there was an increase of 112% (2,500+ instances) of such schemes [16]. Remarkably, hackers are moving away from malware use and focusing on compromised credentials, so 71% of intrusions were "malware-free" in 2022. Most importantly, lateral moves from the first compromised host or credential to another in the same environment – took 84 minutes on average in 2022, down from 98 minutes in 2021 [16].

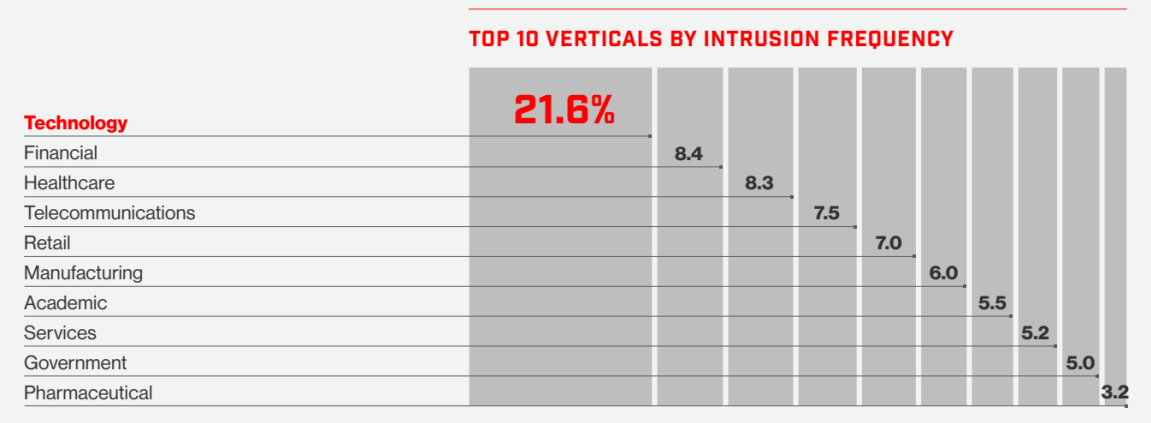


Figure 4.4: Intrusion frequency by industry, copied from source [16].

Figure 4.4 shows the main industries being targeted by attackers. The organization that is the focus of this research is in one of the industries depicted: pharmaceuticals. Recent attacks to the organization followed the pattern described as lateral moves. A phishing email purportedly from an existing information system requested the entering of the credentials in a web page. When one of the victims did enter the username and password, the attacker logged into the email system with the compromised credentials and started sending emails from the compromised account that were not easy to detect as malicious. A preventive technical control - Multi-Factor Authentication (MFA) that forces the user to approve any login on a different system - did not work as the unsuspecting user just approved MFA challenges without thinking that an attacker was taking ownership of the credentials.

The ever growing attack surface can be best measured by metrics around vulnerabilities. A vulnerability can be defined as a flaw in software code that may allow unintended access to information resources. The effort to identify, define and communicate vulnerabilities broadly was started in 1999 [72], achieving remarkable success. The Common Vulnerabilities and Exposures (CVE) dataset started with

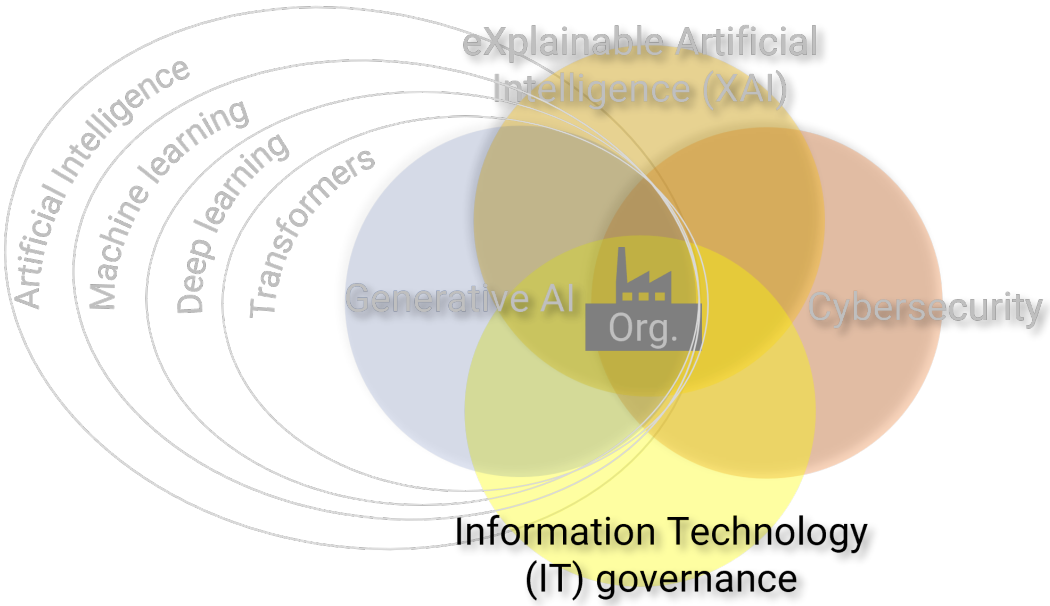


Figure 4.5: Information Technology governance in context, created by E. Lopez.

approximately 900 vulnerabilities, growing into the standard repository containing more than 160,000 definitions used worldwide.

4.3 Information Technology (IT) governance

Information and communications technologies enable many organizational processes across all industries. Myriad information systems are conceptualized, designed and built by multiple stakeholders to meet organizational needs. Modern organizations usually have an Information Technology (IT) department that is accountable for the planning, implementation and support of its information systems. Realizing the benefits from the use of IT, while optimizing risk and resources, requires the implementation of a governance framework [52]. This section articulates the backdrop for the implementation of an IT solution in an organization. In the design science research

framework, it is the environment where the phenomena of interest exists, establishing the fundamental dynamics influencing a successful implementation of the IT artifact.

One of the most widely adopted governance frameworks is issued by the organization Information Systems Audit and Controls Association (ISACA), named COBIT. Beginning with COBIT 2019, the governance included linkages to many other popular business models and standards, effectively creating a holistic governance meta-model that I use in my research. IT governance is formulated and executed in the context of enterprise governance. It is defined as the set of responsibilities and practices providing strategic direction to an organization, ensuring objectives are met while appropriately managing risk and optimally using resources [61]. Two dimensions have been identified in enterprise governance: conformance and performance [54]. Both dimensions should be in balance as enterprise governance acts as the accountability framework for the organization. There are well-established oversight mechanisms such as audits or assurance to validate that good corporate governance standards are being implemented.

IT governance ensures that stakeholder needs are understood, direction is set and performance and conformance are monitored against the articulated objectives. In contrast, management is about the planning, building, operating and monitoring of activities in alignment with the objectives set by the governance processes. For the purposes of this text, and unless otherwise noted, this study refers to governance as the overall umbrella term for both enterprise IT governance and management. The scope for Enterprise Governance of IT (EGIT) includes every part of the organization in which information and technology is used in the pursuit of value – including IT and interfacing functions. Figure 4.6 depicts the different elements in the governance

system: scope, goals, objectives and components.

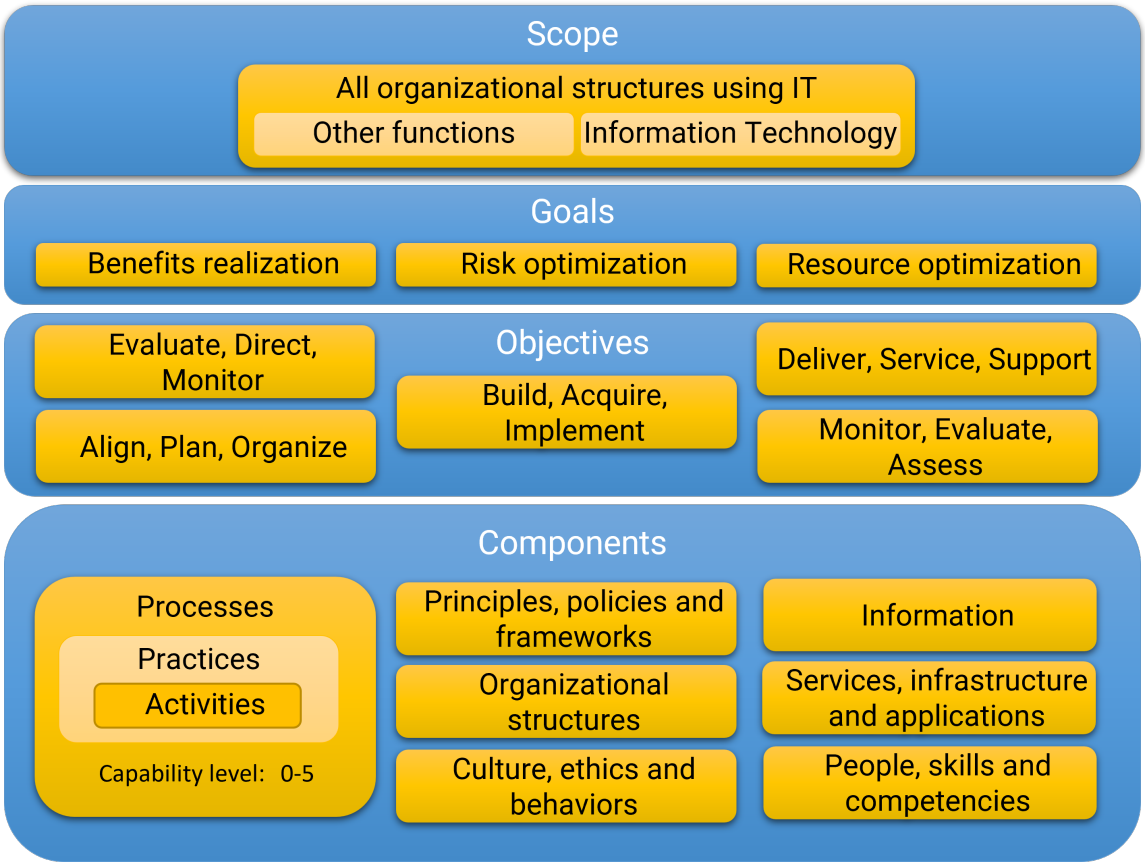


Figure 4.6: COBIT 2019 Enterprise Governance of IT (EGIT) system: scope, goals, objectives and components. Created by E. Lopez, from information in [51].

Scope Enterprise IT governance covers both IT and all other functions or departments that use information systems for the delivery of value. This is a holistic approach where the value of IT is sought after across the organization.

Goals IT exists for the purpose of realizing the benefits from information systems use. This is the primary goal which is supported by optimizing the risk and resources

required. Factors such as the type of organization, geographical footprint and industry influence the exact semantics behind each of these goals. Heavily regulated industries would have a significant focus on the optimization of risk, as is the case in pharmaceuticals with the need to comply with Good Manufacturing Practices [87].

Objectives Multiple objectives are defined for enterprise IT governance in the pursuit of the goals. There are 40 objectives in total, grouped in five domains: Evaluate, Direct and Monitor; Align, Plan and Organize; Build, Acquire and Implement; Deliver, Service and Support; and Monitor, Evaluate and Assess [52]. Figure 4.7 displays the objectives that are part of the COBIT 2019 model [51].

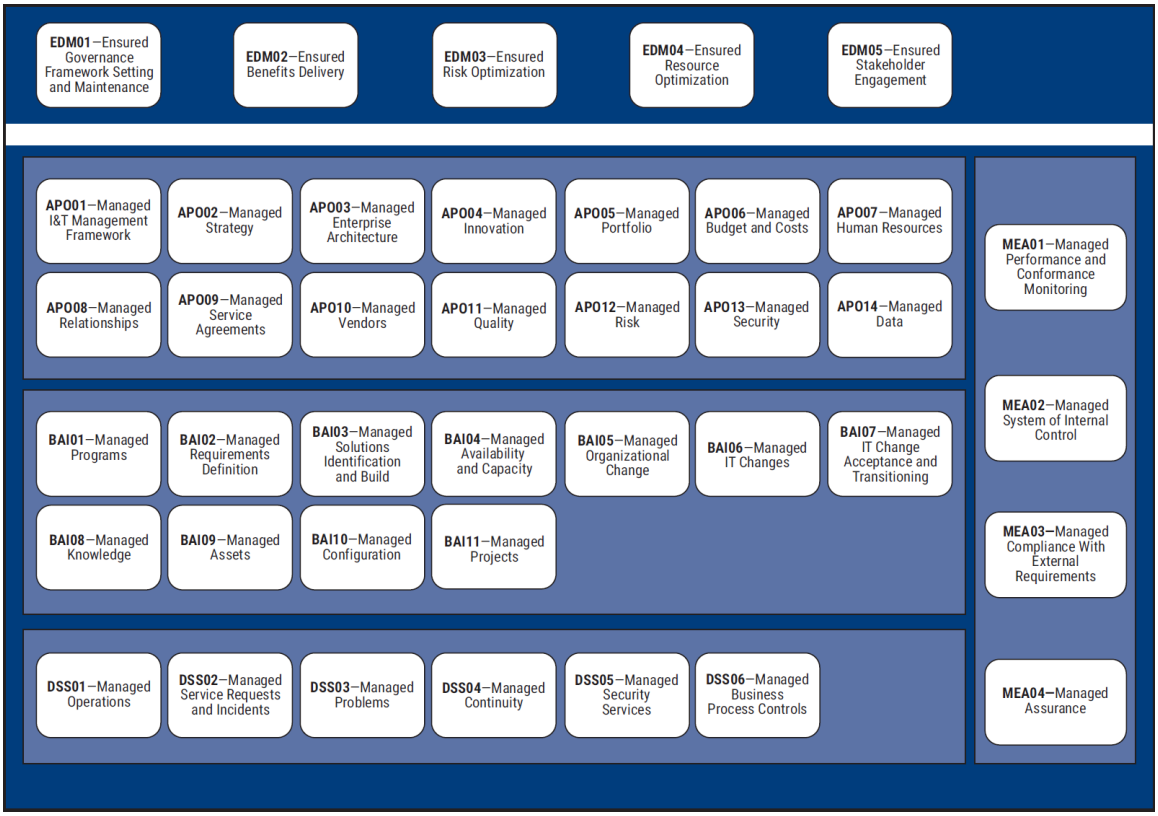


Figure 4.7: Governance and management objectives as per COBIT 2019. Copied from source [51].

Enterprise IT governance components To achieve the objectives, multiple components are used in a governance system. There are seven types of components articulated by COBIT 2019, as follows:

- **Principles, policies and frameworks**: These are the ways in which the organization provides guidance. It is the explicit knowledge (i.e. codified) about how the IT operation runs at the highest level. These components are usually instantiated as controlled documents (such as Standard Operating Procedures) that may subsequently used for employee training and communication.

- **Processes:** 40 processes are defined, each seeking the achievement of an enterprise objective (e.g., the objective "Managed vendors" is proceduralized in the process "Manage vendors"). Each process reaches a capability level (from 0 to 5) that denotes how mature the process operates within the context of the organization. Every process has a set of practices associated with it, composed of a number of activities [50].

- **Organizational structures:** Decision-making dynamics in the organization revolve around people. They are stakeholders organized in business units responsible for the processes in scope [52]. Organizational structures are instantiated in the organization's departments, and documented in organizational charts references in the controlled documentation.

- **Culture, ethics and behavior:** These are critical elements that can be managed towards the design and implementation of a governance system [52].

- **Information** This component type includes knowledge assets used as inputs, or produced by the processes in EGIT. For the purposes of this study, the terms

data, information and knowledge are used interchangeably [50].

- **Services, infrastructure and applications:** This include all technology artifacts used to run the organization [50].
- **People, skills and competencies:** Operating an IT function requires a diverse set of skills at all levels. This type of component distinctly defines the people-centric requirements of EGIT.

Closely related with the semantics of governance, *risk* is defined as the effect of uncertainty on objectives [56]. Although the ISO standard interprets the effect as being potentially negative or positive, in the context of this study the risk effects have a negative connotation. Risk management refers to the coordinated set of activities directing and controlling the organization; including the design, implementation and monitoring of controls – or measures that modify risk [56] Controls – or internal controls, as it is widely known in industry – are an essential component of a governance system in its ability to manage risk and create value [77]. The Sarbanes-Oxley (SOX) United States federal law introduced a set of requirements that public organizations need to comply with, seeking accuracy and transparency in the companies' published financial results. Among the multiple sections describing SOX requirements, section 404 pertains to the responsibility of management for designing, implementing and monitoring/assessing the internal controls in the organization. Typical IT general controls include logical access controls , systems development life cycle controls and change management procedures [77]. Extant literature points to three types of controls [27, 40, 106]:

- **Physical controls** are those that protect the assets from physical actions by a

threat agent. Spatial separation of departments to protect against eavesdropping is a typical example.

- **Technical controls** use technology to achieve the security goals. An access control software is a representative example of this safeguard.

- **Administrative controls** encompass all those safeguards that are procedural, cultural or legal in nature.

A preventive control works towards impeding the realization of the risk, whereas a detective control assumes the risk event has taken place, and manages the response to it [19]. Auditing for compliance assesses the existence, proper design and active use of controls [4, 49].

4.4 Artificial Intelligence

4.4.1 Historical context

Artificial Intelligence (AI) is, arguably, the most important information systems development in the preceding 15 years. Although the AI term was coined as early as 1955 - by John McCarthy and a group of scientists in the first workshop dedicated to the topic [74] - it is in recent years that significant breakthroughs have made it ubiquitous. AI today enables many processes, from relatively trivial applications like games, weather forecasting and chat-bots to essential activities such as medical diagnostics and search.

The pace of innovation in AI is relentless, and appears to be accelerating. Early examples in gaming illustrate the progression. A Bayesian learning system beat elite

Othello players in 1980 [86], but it was perhaps the defeat of chess' world champion Kasparov in 1997 by IBM's Deep Blue that caught humanity's imagination [47]. The approach was akin to brute force: massive computing power capable of evaluating millions of positions per second and using purpose-built chips. A very different technique was used by IBM's Watson in 2011 to beat Brad Rutter and Ken Jennings in *Jeopardy!*, since parallel algorithms were needed to parse keywords and understand the clues [45].

Although gaming certainly provides an interesting backdrop to showcase AI's evolving capabilities, it is another area that best illustrates the possibilities: Natural Language Processing (NLP). On July 2020, the company OpenAI unveiled the largest language model to date, GPT-3 [9]. As an NLP model, GPT-3 was designed to excel at certain particular tasks. But the massive scale of the model produced an unexpected result: strong results for tasks other than the ones the model was trained to do. Fast forward to 2023, and ChatGPT has unleashed a storm of achievements generally referred to as Generative AI (GenAI), where AI systems can create content (text, images, audio) that is very difficult to differentiate from human-created material.

Humanity's desire to "forge the gods" [75] has fueled the creation of artificial entities seeking to replicate intelligence. Although 'artificial' may have pejorative connotations, I choose to interpret it as man-made, produced by art rather than by nature [92]. 'Intelligence', on the other side, is intrinsically related to a natural human attribute and the ultimate objective for artifacts to understand, learn and perform *any* intellectual task is usually referred to as Artificial General Intelligence (AGI) or strong AI – and it is considered to be out of reach at this time. In contrast, weak

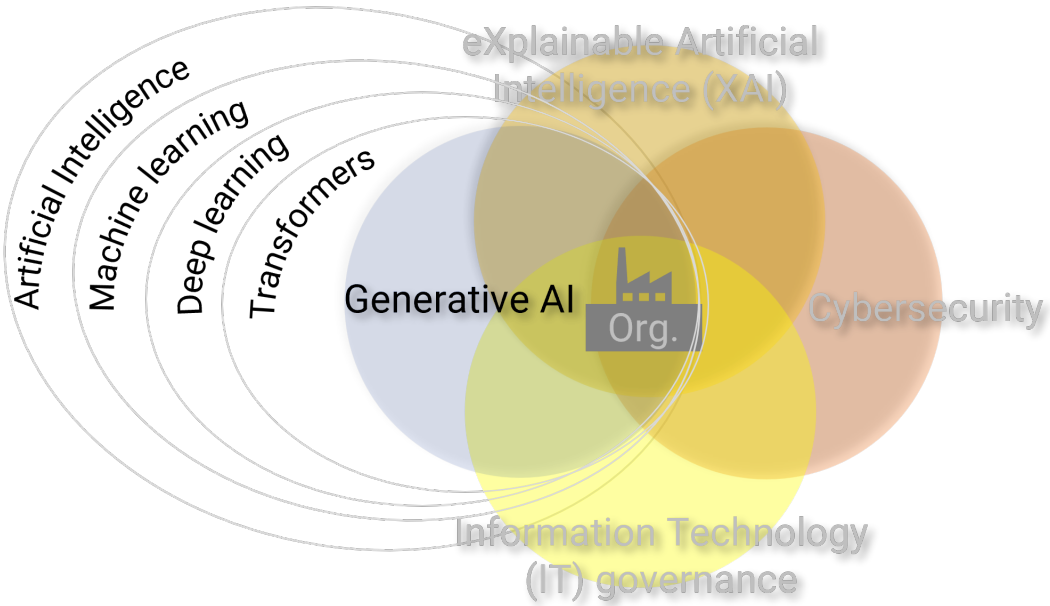


Figure 4.8: From Artificial Intelligence to Transformers. Created by E. Lopez.

or narrow AI, or just AI, is usually focused on a single task or activity – with the implicit objective of matching or surpassing human performance.

This section delves into the specific advances that have led to the current state of the art: Transformers [93, 94]. Figure 4.8 depicts the literature review approach.

4.4.2 Machine Learning

Information systems have revolutionized virtually every facet of human endeavours. How these systems are created has also evolved significantly, departing from hardware-only wiring towards low-level languages such as assembly, and culminating with very expressive higher-level program languages that can create sophisticated representations of reality. Machine learning changes this computing paradigm – rather than

building information systems by explicitly coding rules (i.e., declarative programming) the software artifact is created so it can learn from examples (be 'trained'), which is closer to imperative programming. More formally, machine learning is programming computers to optimize a criterion using example data [26]. In machine learning there are relatively few literal instructions telling the artifact what to do, replacing it with instructions about what variables to optimize during a "training" phase.

Arguably the simplest machine learning algorithm is a linear regression. The system is provided with the data for finding the parameters (i.e., coefficients) that reduce the error between a linear function and the data. At the end of the training stage, the parameters found can be used for predicting new data. There are many algorithms that could be considered machine learning. Figure 4.9 depicts a taxonomy for machine learning algorithms capturing some of the most popular in use today.

As can be observed in the depiction, the versatility of machine learning is remarkable. The basic tasks that can can use machine learning are prediction and classification, but there are many other applications where finding patterns in the data can offer value. Given that the most recent advances evolved from Artificial Neural Network (ANN) and towards Deep Learning, I proceed to explain their mechanics and how they evolved to the state of the art as it is known today.

4.4.3 Deep Learning

The departing point for the Deep Learning revolution can be traced back to Artificial Neural Networks, or ANN. In their simplest form, ANN can be conceptualized as

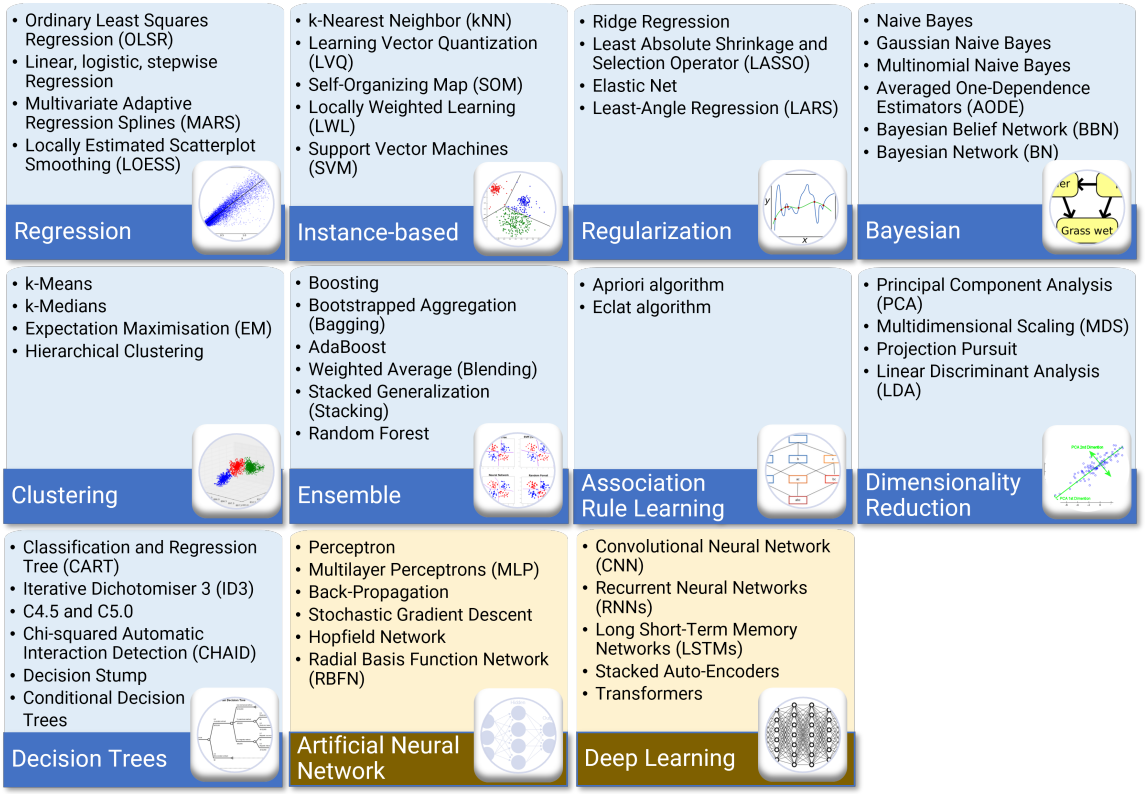


Figure 4.9: Machine Learning by algorithm type. Created by E. Lopez with information collected from [28, 71].

mathematical models composed of 'neurons', where each neuron performs mathematical operations depending on a set of parameters. Figure 4.10 depicts a single neuron and how it fits within a neural network architecture.

The neuron depicted shows how two data inputs x_a and x_b – which we will continue referring to as *features* – are multiplied by the parameters w_1 and w_2, added a constant value b and entered into an activation function σ. The objective of the activation function is to add non-linearity to the calculation, which is essential for certain machine learning tasks such as classification. The calculated value resulting from the transformation is denoted as y.

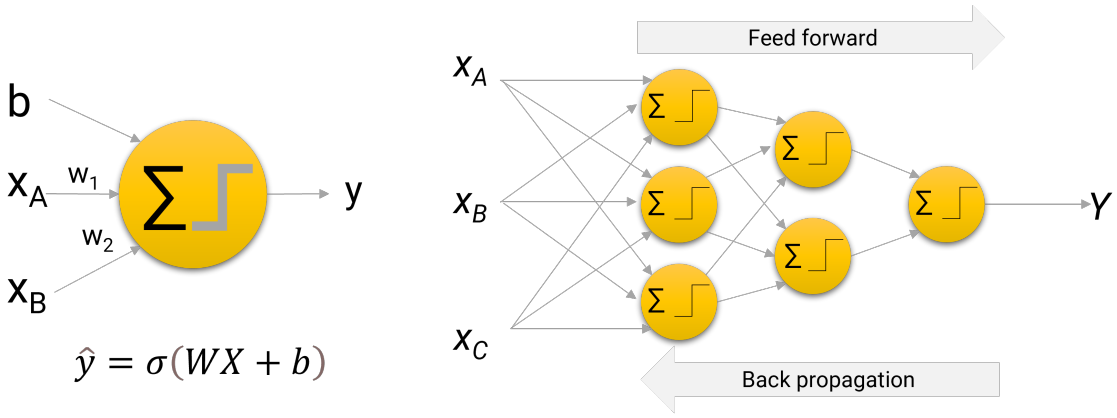


Figure 4.10: Depiction of one neuron and a simple neural network, created by E. Lopez.

Using an existing dataset with feature values x_a, x_b and y, an algorithm will estimate how different a predicted $\hat{y}$ is from the actual value y. This difference is quantified with a cost function J defined as:

$$J(w, b) = \frac{1}{m} * \sum_{i=1}^{m} L(\hat{y}_i, y_i) \tag{4.4.1}$$

Where m is the total number of observations in the dataset, w are the parameters (i.e., weights) and L is the cross-entropy loss function for sample i. The processes by which the neuron is trained and the optimal parameters w are found are referred to as *feed forward* and *back-propagation* [78]. The right side of Figure 4.10 shows how multiple neurons can be used to create a network that follows the same principles described.

Neural networks are very effective function approximators when given a suitably large training dataset [62]. The samples are entered in *batches* with a full feed-forward and back-propagation cycle referred to as an *epoch*. The batch size, number of epochs and other parameters that control how the learning occurs are denoted

hyper-parameters. Once the learning has been completed, i.e., the parameters w estimated, the network can be used for analysis of new data not seen previously by the model.

In prediction, the value y is a number – such as predicting the temperature or the price of a house. In contrast, classification is about finding the class that the new data belongs to, so y would yield a value from a predefined group. A typical application happens in fraud detection, where a binary classification ($y = 0$ or $y = 1$) identifies records as normal or anomalous. If the dataset includes labeled data (i.e., the class that each sample belongs to) the learning is *supervised*. Conversely, if the data used for the training is not labeled, the training of the model is referred to as *unsupervised* learning.

As multiple layers of neurons are added, the complexity of the model increases dramatically. The biggest challenge was the "vanishing" and the "exploding" gradient problems, where networks with a large number of hidden layers ('deep networks') did not train properly because the signal disappeared or exploded [109]. Advances in hardware such as the creation Graphical Processing Units (GPU) powered ways to perform back-propagation that avoided the gradient issues, allowing the use of 'deeper' networks.

Deep learning can be defined formally as computational models with many layers that learn multiple representations of the data at varying levels of abstraction [63]. Deep learning is able to find very complex patterns in many different types of data (e.g., sequential) such as speech, text or video. Deep learning is more effective at using raw data directly, without the careful crafting of the features demanded by the existing machine learning techniques [31].

One of the areas where deep learning has shown its power for representation learning is language processing. The standard technique used at the time was based on statistical modelling of short sequences, for example the number of times a given word appeared in a sequence of length N – referred to as N-grams. Deep learning was able to learn distributed representations of the words, enabling generalization to new sequences not seeing in the training data. A remarkable example of this generalization is word vectors, sometimes referred to as word embeddings.

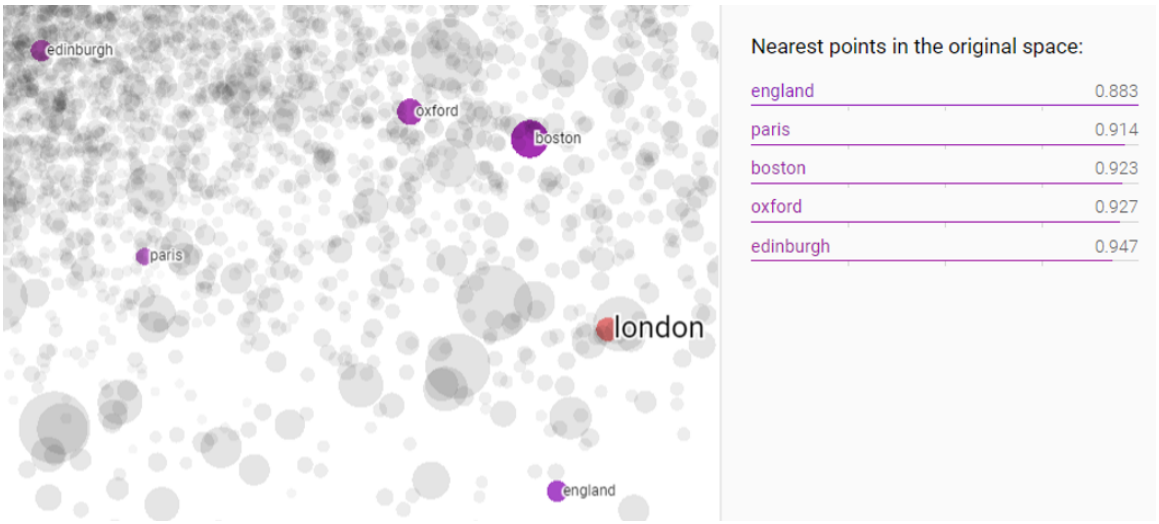


Figure 4.11: Word *London* in Word2Vec: reduced to a 3-dimensional space using Principal Component Analysis (PCA). Created by E. Lopez using https://projector.tensorflow.org.

A word embedding can be calculated by a multi-layer neural network with the task of predicting the next word in a sequence or context. This is an unsupervised task where the training sequences are obtained from a set of documents where each word is assigned an integer number or index, usually codified as a one-hot vector. Once training is finalized, the parameters w can be used as the distributed representation of the words – features that are not mutually exclusive that represent different

dimensions for any given word. An example of the algorithm is posited in the seminal work *Distributed representations of words and phrases and their compositionality* [76], referred to as Word2Vec. The algorithm calculates the dense embedding vectors (200-dimensional) for 71,291 words based on the United States Google News corpus. The resulting numeric vectors can be used for mathematical calculations, including similarity as measured by distance.

Figure 4.11 shows the word *London* along with the 5 closest vectors in the 200-dimensional space. As can be observed, the closest word to *London* is *England*, shortly followed by *Paris* and *Boston*. Based on the corpus used, and without providing explicit semantics associated with the calculation of the word embeddings, Word2Vec found that *London* and *England* are closely related, while also being conceptually closed to two other cities.

It is important to note that there are implicit biases that permeate the meaningful vectors found. An example is illustrated in Figure 4.12. The word *assassin* has as second closest point in the hyperspace the word *Oswald*, the individual that assassinated John F. Kennedy in one of the most notorious events that have taken place in the United States. The Word2Vec algorithm found the relationship between the two words based on the corpus from which it was trained. This is a prime example of bias inherent in the dataset that can be found across many machine learning applications nowadays.

Deep learning proved suitable for identifying representations for sequential data. Specialized neural networks such as Recurrent Neural Network (RNN) were formulated. Each state data representation – referred to as *hidden state* h_t includes information gathered when the earlier elements of the sequence were processed [12]. The

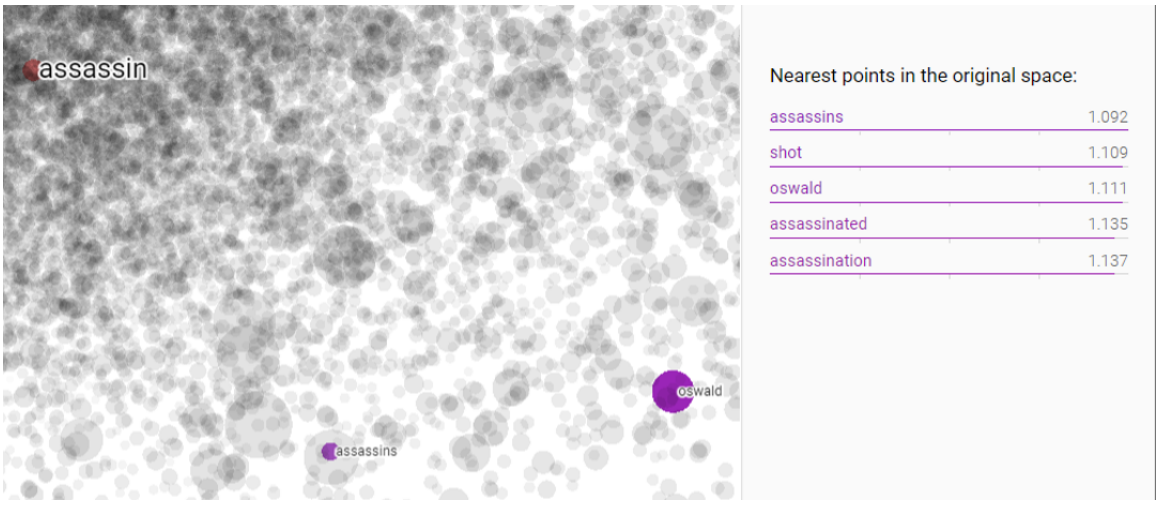


Figure 4.12: Word *Assassin* in Word2Vec: reduced to a 3-dimensional space using Principal Component Analysis (PCA). Created by E. Lopez using https://projector.tensorflow.org.

advantages brought by RNN architectures were offset by a fundamental limitation: the ability to 'remember' what took place early in the sequence. To overcome this limitation, the Long Short-Term Memory (LSTM) architecture was designed. LSTM neurons use mechanisms, or gates, for regulating the flow of information that is processed sequentially [91]. The gates permit the LSTM to learn during training what data in the sequence is important or is discarded. Figure 4.13 shows an example of the dynamics generated with LSTM.

A typical application where LSTM excels at is the prediction of the next word in a sequence based on its context (adjacent words) - typically referred to as a language model. During training the LSTM model learns what are the typical words that would follow a given sequence of words. The implementation in Figure 4.13 illustrates an encoder-decoder architecture, where the encoder outputs the representation of the sequence which is then entered into a decoder based on a SoftMax activation that outputs the probability distribution of the next words. The illustration shows the

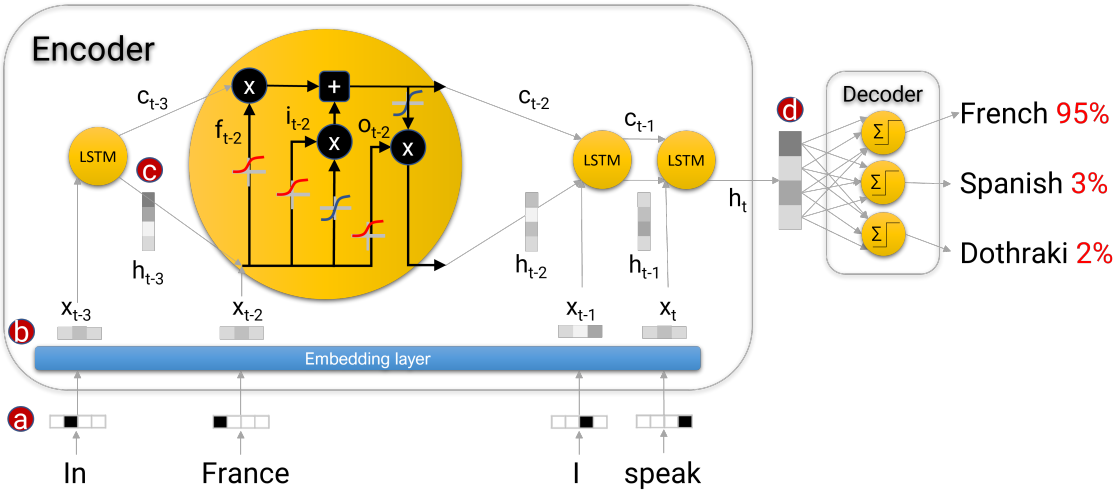


Figure 4.13: Long Short Term Memory - gates and operations. Created by E. Lopez.

processing of a single LSTM unit logically 'unrolled' to explain what happens within it. First, the original raw data (a sequence of words) is indexed and each token represented by a one-hot encoded vector, as captured in 4.13.a. The actual code assigned to a word is not important as it does not carry any meaning to be used in the analysis. The data is then processed through an embedding layer obtaining a numerical representation of the word as a dense vector, illustrated as segment 4.13.b in the image.

The LSTM model processes sequentially each word, calculating the hidden state as a new dense vector representing the sentence up to that word. The embedding for the x_{t-3} word 'In' is processed first, with two values produced: the cell state c_{t-3} and hidden state h_{t-3}, identified as 4.13.c. The LSTM then takes the next word x_{t-2}, the preceding cell and hidden states and obtains the new cell state c_{t-2} and hidden state h_{t-2}. The process continues until all the words in the sequence have been transformed in the LSTM and produce the final hidden state or encoded sequence h_t, identified as 4.13.d in the figure.

The mechanisms that an LSTM uses for processing are denominated gates, calculated using matrix operations that are optimized in hardware such as GPU. The first gate – known as the 'forget' gate – for time *t-2* in the illustration is calculated as follows:

$$f_{t-2} = \sigma(W_f.[h_{t-3}, x_{t-2}] + b_f) \tag{4.4.2}$$

Where W_f are the learned weights, h_{t-3} is the value of the preceding hidden state, x_{t-2} is the word being processed, b_f is the bias and c_{t-3} is the value of the preceding cell state . The activation used in the forget gate is through the σ function. The following is the 'input' gate calculation tor *t-2*:

$$i_{t-2} = \sigma(W_i.[h_{t-3}, x_{t-2}] + b_i) \tag{4.4.3}$$

Where W_i are learned weights, h_{t-3} is the preceding hidden state and x_{t-2} is the word being processed. Using the outputs from the input and forget gates the new cell state is calculated as follows:

$$c_t = i_{t-2} * tanh(W_c[h_{t-3}, x_{t-2}] + b_c) + f_{t-2} * c_{t-3} \tag{4.4.4}$$

Where W_c are learned weights, h_{t-3} is the preceding hidden state and x_{t-2} is the word being processed. The output gate is calculated as:

$$o_{t-2} = \sigma(W_o.[h_{t-3}, x_{t-2}] + b_o) \tag{4.4.5}$$

With the final hidden state h_{t-2} calculated as:

$$h_{t-2} = o_{t-2} * tanh(c_t) \tag{4.4.6}$$

During the training stage all the parameters $W$are learned based on the training dataset. After a suitable number of epochs, the optimal parameters value can then be used for prediction activities.

Albeit an improvement over RNN, LSTM still have issues when dependencies exist in very long sequences. In addition to this, LSTM are train slowly due to its sequential processing. The search for a more optimal solution that could address these issues resulted in the concept of *attention*.

4.4.4 Transformers

Attention was first mentioned in 2015 [11] with the seminal paper on transformers published by Vaswani et al. in 2017 [102]. The original transformers architecture follows the encoder-decoder structure calculating data representations using the concept of *attention*.

As it is the case with LSTM, a Natural Language Processing (NLP) example is a suitable way to develop an intuition for the goal of attention. The phrase "***Transformers*** *are great with long sequences because* ***they*** *use self-attention*" has an implied dependency between 'Transformers' and 'they'. Although this relationship is – arguably – evident for a human being, for an artifact the word 'they' may be referring to 'long sequences'. Attention focuses on formulating a quantifiable metric for this dependency in a language model. We will use Figure 4.14 to describe the dynamics.

The Transformer's fundamental architecture is that of an encoder/decoder. It

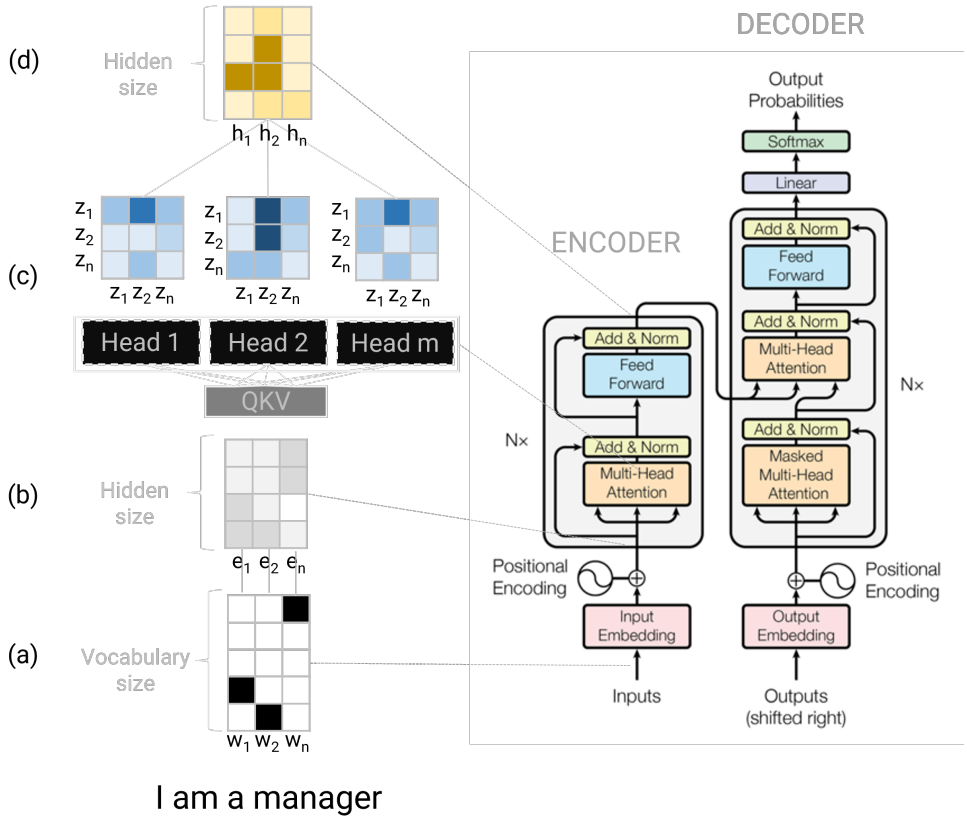


Figure 4.14: Encoder representations in the transformers architecture. Created by E. Lopez using an image from [102].

takes an input to the encoder and that outputs a representation of the data. The decoder then takes the output and in the decoder it reconstructs it in a recurrent manner. We now proceed to explain the process in more detail.

Encoder

In a similar fashion to the representations found through word embeddings, the first stage is the transformation of sequences into numeric formats that can be processed by the neural network layers. Given a sequence of words w_1 to w_n in a sentence, the one-hot representation – figure 4.14(a) – is entered in a input embedding layer,

with positional encoding information added before the encoder stack. In contrast to the word embeddings explained previously, the position of each word needs to be captured explicitly. An algorithm such as Word2Vec or an embedding layer in an LSTM model has already positional information for each token (i.e., word) since those are processed in strict sequence. An algorithm such as Word2Vec assigns a unique embedding representation for each word irrespective of where it is used. In the phrase "I go to the river **bank**" the embedding for bank would be the same as in the phrase "I made a deposit in the **bank**". In contrast, the word embeddings for the same word in different contexts, i.e., with different neighboring words, are different when using attention as it is the case with Transformers.

The embeddings e_1 to e_n for the sequence of words are depicted in figure 4.14(b). The size of the hidden state (i.e.,embeddings) is a configurable hyperparameter of the model, also knows as the 'hidden size'. The typical number of dimensions found in transformers' implementations range from 768 to 2048. The resulting vectors for each word are then multiplied by three weight matrices W^Q, W^K and W^V that will be learned during training resulting in three matrix abstractions: Query (Q), Key (K) and Value (V). To quantify the relationship between every word pair, the calculation that takes place is shown in equation 4.4.7.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4.4.7}$$

The dot product of Q and K is scaled by the square root of the dimension of the K matrix. A Softmax transformation is then applied (producing a matrix with values from 0 to 1) and then multiplied by the V matrix to produce the attention matrix. The calculation of attention underpins a radical-yet-simple approach to quantifying

relationships between words. It is calculated in a parallel manner, with all the tokens in the sequence being inputted into the model. Furthermore, there are multiple *heads* or independent attentions calculated, allowing the learning of representations in different subspaces. We can use again language to develop the intuition: each head is randomly initialized at training time, and will find the parameters (i.e., weights) for the different types of relationships such as syntactic, grammatical or gender between any two words.

The outputs from each head are square matrices with dimensionality depending on the number of words in the sequence as it is shown in figure 4.14(d). The final step in the encoding exercise is to aggregate the insights from each head achieving a single representation for each word in the sequence, illustrated in figure 4.14(d) for words h_1 to h_n. The output for the first encoder may be entered on a second encoder in the stack where similar calculations take place. The process is repeated as many times as there are encoders in the model.

In summary, the encoding process receives a number of words, and outputs a matrix that represents a deeper meaning of the input: an abstraction that is highly contextualized. For example, in the case of a Transformer used in language translation, the input can be a sentence in English (e.g., "I am a manager") and the output is the French translation (e.g., "Je suis directeur"). During the training phase, both sentences are known, with the encoder receiving "I am a manager" and outputting a meaningful vector representation.

Decoder

The decoding functionality in the transformer is similar to that of the encoder. The same approach used for findings the embedding - including the positional encoder - is utilized as the data entry to the decoder. This is represented in 4.15 (a) and (b).

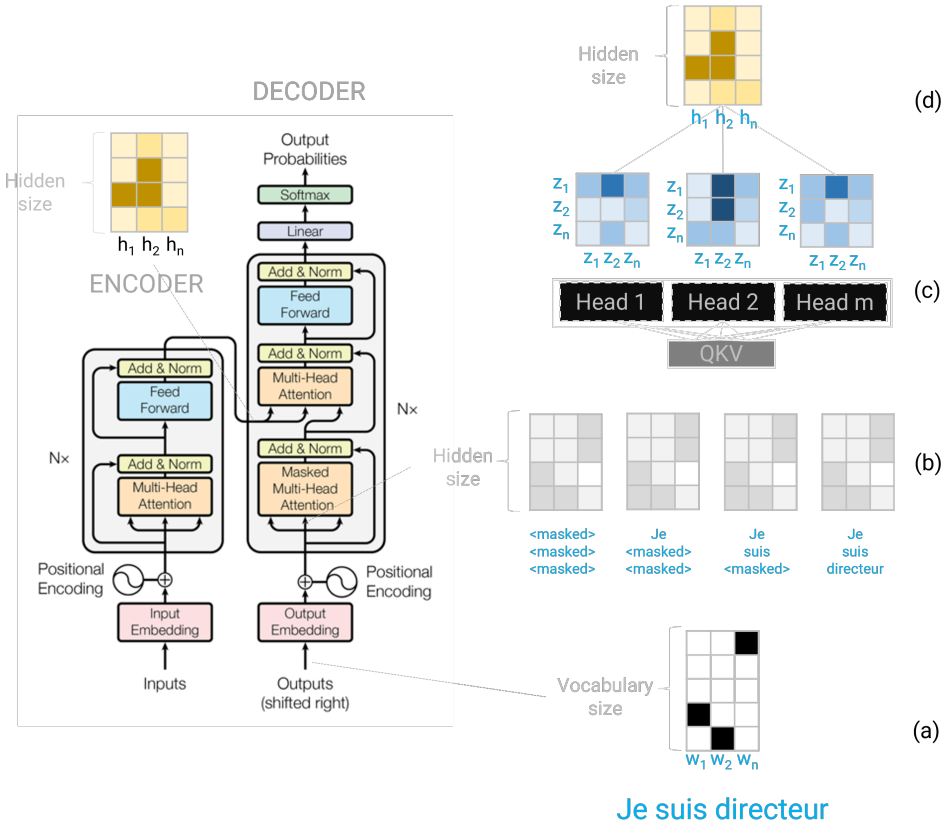


Figure 4.15: Decoder representations in the transformers architecture. Created by E. Lopez using an image from [102].

In the case of training the model for translation, "Je suis directeur" embeddings are found, and positional encoding used. Then, masking elements of the output, the model is trained to find the QKV values that are able to reconstruct "Je suis directeur" from "I am a manager".

4.4.5 State of the art in Transformers: towards Generative AI

The evolution in Transformers has been remarkable. The speed by which innovative architectures are trained and deployed makes any effort to establish "state of the art", to some degree, meaningless. The number of different models is too high to capture and ever growing, but two have been fundamental in the development of this research - and are explained next.

BERT: Bi-directional Encoder Representation from Transformers

These are models that use only the encoder elements in the transformer architecture [20]. BERT models process datasets in two distinct stages. The first one is denominated pre-training, where unsupervised learning takes place through two tasks: a Masked Language Model (MLM) and a Next Sentence Prediction (NSP). In MLM, a number of words are masked in the training corpus with the goal of predicting the hidden word. In the second task, the model is trained to predict a given sentence based on the preceding one. At the end of the pre-training task, a language model has been learned that can be used for many other tasks or further trained or updated during a fine-tuning stage.

The masked language model learned during training in BERT can be effectively transferred to other adjacent tasks using the same data. The transfer learning possible with BERT enables myriad practical applications for any domain where attention between tokens (i.e., words) in a sequence follows complex patterns. Pre-training of a BERT language model requires significant resources, but once it has been completed for the first time, it can be fine-tuned efficiently in order to keep it current.

BERT has proven to be very popular, and myriad derivatives and variants have been published, but were limited to encoder-centric tasks. This meant that BERT was not used for generation of content, which was the remarkable development in the preceding year, as explained next.

GPT: Generative Pretraining Transformer

Generative Pretraining had been explored by George Hinton in 2012 [44], before *attention* was developed. Once the Transformers architecture became the focus of research activity, the company OpenAI used it for their creation of the first model, called GPT-1 [85]. The training dataset was BookCorpus, which contains long passages of text suitable for learning natural language patterns and is approximately 3GB of text. GPT-2 shortly followed, trained on approximately 40GB of text and producing a model with 1.5 billion parameters.

GPT-3 was published by OpenAI in July of 2020, trained on 570GB of data producing a 175 billion parameter model. GPT-3 was fine-tuned for user interactions, resulting in the very famous ChatGPT released in November 2022 [55]. The excitement around ChatGPT has been remarkable. It was a spin-off from a previous OpenAI model, enhancing it with Reinforcement Learning from Human Feedback (RLHF), dramatically improving previous results[79].

In March 14, 2023 OpenAI published GPT-4. There are very few details shared by OpenAI and Microsoft (which acquired a large part of the company), but it is estimated that it is a model with approximately 1.7 trillion parameters, trained with RLHF [32].

Towards foundation models

The dramatic success of BERT and GPT can be best contextualized by using a term coined in July 2022: Foundation Models [8]. It is defined as a model that has been trained on broad data, usually very-large scale, and that can be adapted to other tasks by a method such as fine-tuning. Foundation Models, as described by its authors, are the byproduct of emergence and homogenization.

As was explained previously, AI devolved into functionality being induced from data, as opposed to be declaratively programmed. Models such as GPT-4 fundamentally understand patterns in language, as induced from the gigantic data used in their training, and can be fine-tuned for other tasks. It epitomizes the concept of *transfer learning*, where the model is trained on a surrogate task during pretraining, and then adapted to other via the fine-tuning.

The architecture generalization, or *homogenization*, can be gleaned from the sheer number of models that have spawn over the preceding years - vastly using the Transformer and the concept of attention. The same fundamental architecture is used for Natural Language Processing (NLP), vision and applications as diverse as speech or protein sequencing.

The very significant scale of the AI models trained recently has led to new insights that may point towards fascinating research streams. Machine learning models fundamentally try to find a way to generalization. The ultimate objective is for an ML model to be able to perform its task with a high level of accuracy using data that has not been seen previously. Generalization can be explained in the context of training and test errors in a model. Training a model with data reduces the training loss while finding the parameters. However, common practice has been that achieving

minimal training loss may be counterproductive since it can lead to over-fitting: a model with a high test error. Recent papers have shown a different scenario taking place. Overparameterized models – those with more parameters than training data – appear to eventually reduce the test error even though the training error has been fully minimized [39].

The implications of this are very significant. Very large datasets may be more important to generalization than model-specific training [57]. If this assumption is correct, models should be trained first and foremost for generalization. Construct a model that solves several "simple" problems, perform supervised learning with massive datasets and then use it on the specific task.

A follow-up essay from the "Just ask for generalization" author further argues that generalization is equivalent to language. This provocative thought is developed by noting that language (tokens related to one another, usually in a temporal sequence) offers several generalization types that can be applied to other problems [99]. This is perhaps best demonstrated by [69] where a Transformer pretrained on natural language generalizes to scenarios other than language with minimal fine-tuning.

4.5 eXplainable Artificial Intelligence

The notable results achieved with machine learning are somewhat dimmed by a significant drawback: the difficulty in understanding how results are achieved. This section of the literature review delves into the key conceptual elements explored. Some models – such as deep learning ones – can be only understood in terms of their inputs and outputs. Their internal mechanics are not interpretable, a scenario commonly

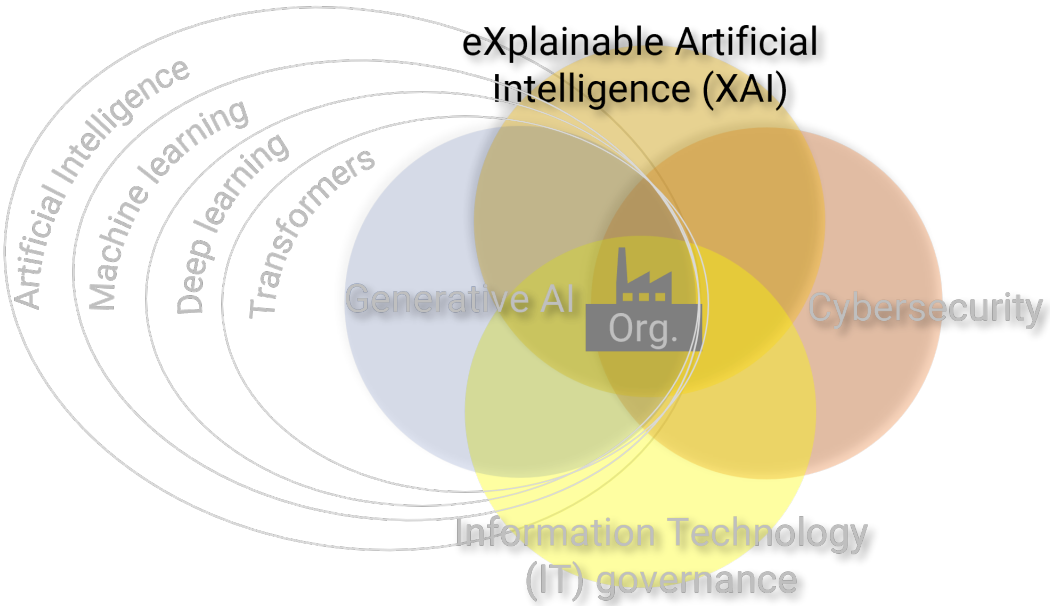


Figure 4.16: Explainable AI for the organization. Created by E. Lopez.

referred to as a black box [10, 110]. The initial approaches to AI involved expert systems and rule-based models, thus allowing humans to understand how predictions or actions by a model were made [13]. However, the complexity of current architectures based on neural networks have a high level of opacity in their mechanics. Notwithstanding the relative simplicity of a single neuron used in Transformers, there is no real possibility for a human - or humanity for that matter - to understand what each of the 1.7 trillion parameters in GPT-4 does.

A good illustration of the interpretability challenges can be seen in cybersecurity. A security analyst belonging to an IT department is responsible for maintaining the confidentiality, integrity and availability of information assets and associated systems. Among the multiple business processes that are usually employed to safeguard the organizations' information, one of the most critical is the periodic review of information systems logs [51]. Logs are usually very large unstructured files where events

taking place in the information systems are recorded. Deep learning can be used to mine these logs thanks to its powerful capability to model sequential data. However, deep learning models are beyond the cognitive reach for any human due to the huge parametric space.

I leverage the definition by [38] for eXplainable Artificial Intelligence (XAI): *... is the use of techniques in Artificial Intelligence (AI) for producing interpretable results that humans can understand and trust.* Although XAI is fundamentally about technical approaches, it must consider the audience for its processes - humans using their perceptions for decision-making.

The relationship with trustworthiness surfaces explicitly as a key objective sought by XAI. At the regulatory level, the European Commission (EC) went as far as publishing ethical guidelines for trustworthy AI [43]. In it, the EC conveys that AI must be lawful, robust and ethical – upholding the principles of *respect for human autonomy*, *prevention of harm*, *fairness* and *explainability*. From the principles, specific requirements are articulated, with assessment items published for ensuring an ethical backdrop for AI creation. Figure 4.17 illustrates the concepts.

The AI requirements listed add additional texture to the XAI dialogue. As was explained previously, AI models based on deep learning are not explicitly programmed. In this case AI is closer to a learning artifact that will establish its rules for prediction and classification based on the data used during training. This dependency not only drives not only the need to maintain privacy with the data used during training – so the model does not publish inadvertently private, sensitive or confidential data – but an understanding on potential biases that are embedded in the training datasets used. XAI techniques can be categorized in two groups: models that are interpretable

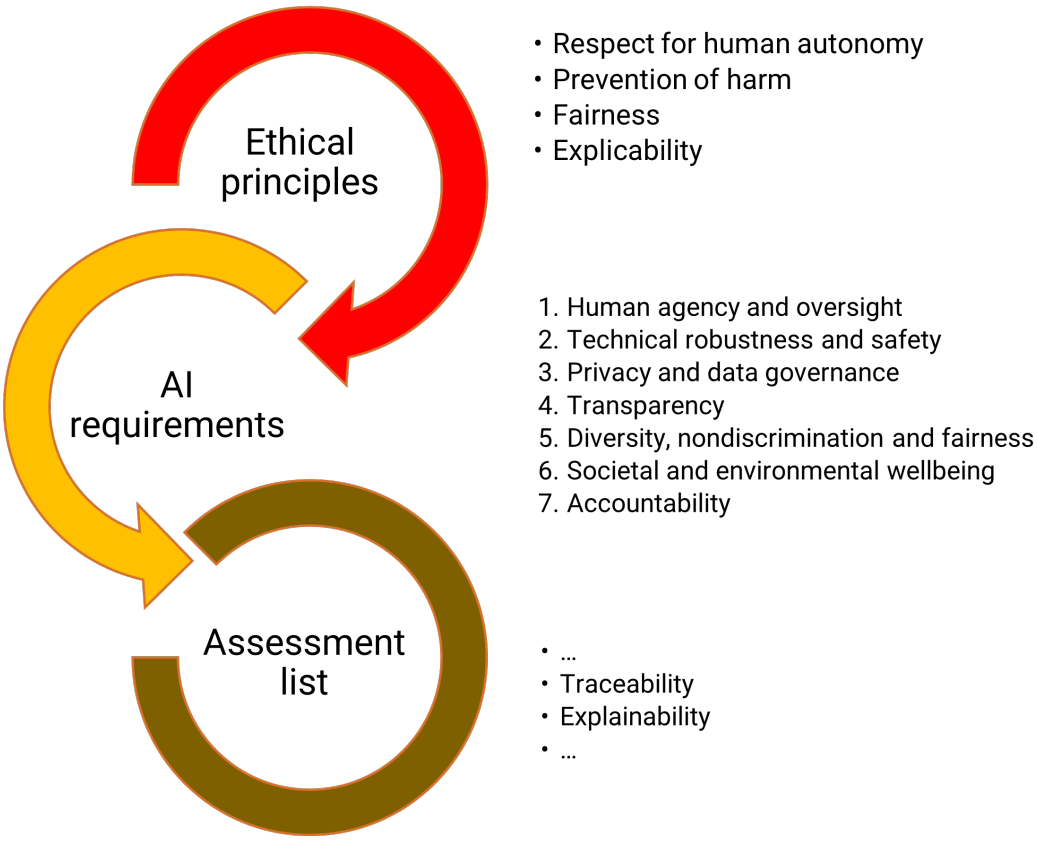


Figure 4.17: Ethics guidelines for trustworthy AI. Created by E. Lopez from information in [43].

by design [13], and models that are not transparent, requiring further action for interpretability (post-hoc explainability) [6]. This taxonomy is depicted in Figure 4.18 shows a taxonomy for the XAI techniques.

4.5.1 Interpretable by design

The attribute of transparency can be assigned at different granularity levels. When it is assessed at the complete model level, it is known as *simulatability*. Similarly, transparency when referring to individual elements in the model is called *decomposability*.

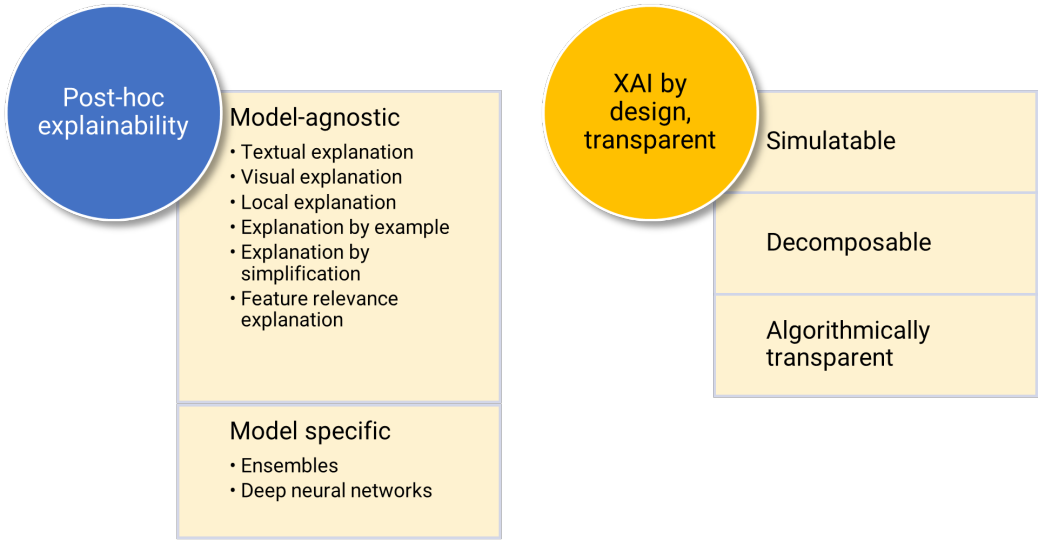


Figure 4.18: XAI techniques taxonomy. Created by E. Lopez from information in [6].

When it pertains to the algorithm, it is referred to as *algorithmic transparency* [110].

Simulatability

For a model to be considered *simulatable*, a human shall be able to simulate its inner workings. It is important to note that rule-based models, although simple to read, can become very complex as the scale increases. Conversely, a neural network with a few units may be considered simulatable because of its relative simplicity.

Decomposability

Transparency at the component level means that the model is decomposable. The case for decomposability is especially interesting in the case of engineered features. The word embeddings concepts explored before shed light over a fundamental element in

machine learning: feature engineering. We formally define a feature as a variable that represent aspects or dimensions of a data instance. The existence of a large dataset does not necessarily suffice for its use in computational and mathematical models [22]. The activities pertaining to the creation, extraction or selection of features from raw data is referred to as feature engineering [23].

Under ideal conditions features are not only suitable for use in machine learning but also interpretable to practitioners. A machine learning model to predict the weather may use as features the historical values for temperature, atmospheric pressure, altitude and wind speed. In many cases these variables are readily available and require no transformation. An additional advantage of this scenario is that the features are meaningful on their own, and fully understandable by human beings, thus achieving decomposability. In contrast, there may be features that may be obtained as a prerequisite to machine learning processes that are not interpretable on their own. We use Figure 4.19 to illustrate the situation. There are four words in the example: *Man*, *Woman*, *King* and *Queen*. Each of these words is contained multiple times in a large corpus of data. The process to engineer the features start with the assignment of an index number (i.e., integer) to each word. The value used does not have any meaning, it is just an identifier for the word within the confines of the model. If the index is sorted alphabetically, the word *King* (1348) is before the word *Man* (2203). An algorithm such as Word2Vec transforms the index to a sparse vector performing one-hot encoding. In the case of *Man* (2203), the representation is a large vector with zeroes (0) in all dimensions but the position 2203, where it is a one (1) as it is depicted in Figure 4.19. The final stage shows the 4-dimensional word embeddings calculated by Word2Vec. Unlike transparent interpretable features, there is no

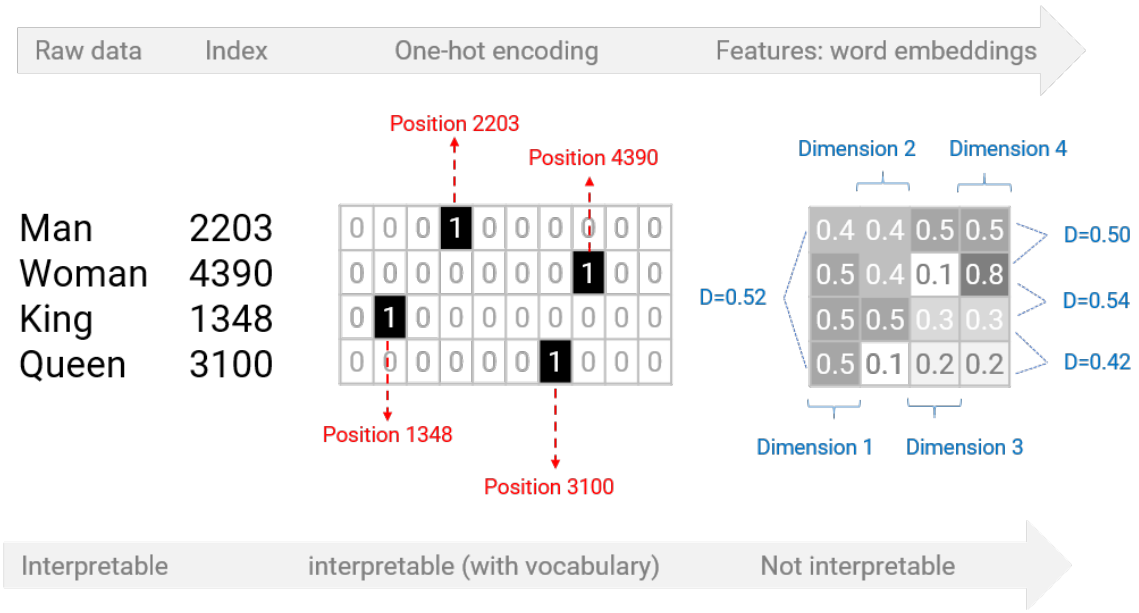


Figure 4.19: Feature engineering for language: interpretability of word embeddings. Created by E. Lopez.

explicit meaning for each of the dimensions. While the index and one-hot encoding can be easily interpreted by consulting the vocabulary, the values found by Word2Vec for each dimension represent undefined facets of the word subject to interpretation by a human being.

The features calculated from the embeddings enable key analytical and computational tasks that are unrelated with dimensions' meaning. As the raw data gets transformed into features, the data becomes less human-interpretable but more usable by AI. This is illustrated on the right side of Figure 4.19. The euclidean distance between the word embeddings for *Man* and *Woman* is less than that between *Woman* and *King*, whereas the words *King* and *Queen* are quite close. In its original form, raw data stored in information systems – i.e., a group of bits that are different between words – is unsuitable for this type of analysis, while the embeddings calculations can

power a new level of understanding. The embeddings concepts can be scaled hierarchically in natural language. It is possible to calculate embeddings for sentences (i.e., groups of words) or documents (i.e., groups of sentences), allowing hierarchical representation of semantics and structure.

Notwithstanding the need and significant value from the use of embeddings, they are not decomposable on their own and therefore require additional effort to ensure interpretability of the model.

Algorithmic transparency

Transparency can also be evaluated at the algorithm level. Many of the algorithms depicted in Figure 4.9 are inherently transparent. Regressions, Bayesian networks or decision trees can be understood with relative ease. However, dimensionality reductors (such as PCA or embeddings) or deep learning networks do not allow this same level of understanding. Deep structures in models such as deep learning can be implemented with few lines of code that abstract the massive complexity behind the calculations. In addition to this, the parameters calculated during the training of the model could also be considered an algorithm created by the algorithm - and thus exhibiting significant algorithmic opacity.

4.5.2 Post-hoc interpretability

Models that are not interpretable by design require a different approach. Post-hoc interpretability techniques can be divided into two groups: those that are independent of the machine learning model used, and those that are particular to a specific model.

Model agnostic

Many model-agnostic techniques gravitate towards ensemble approaches based on simplification. Model simplifications are akin to rule extraction techniques [6]. The objective is to generate explanations after the machine learning algorithm has finished training and/or execution. Some of the model-agnostic techniques include:

Text explanations: Humans employ language to explain phenomena every day. It is, therefore, not surprising to find that one of the options is to generate textual explanation of how a machine learning model works. This may take the form of a core model performing its task, such as predicting or classifying, with a secondary model generating text using machine learning such as recurrent neural networks [110].

Visualization Perhaps one of the most common techniques is to use images for the purposes of interpretability [58, 65, 96]. Visualization can be effective at eliciting understanding by demonstrating regularities or patterns that would otherwise be difficult to grasp.

Local explanations Understanding a black box model globally may be impractical, but sometimes it is possible to understand specific instances, generating trust in the model. An analogy with Netflix may put this technique in context. The recommendation algorithm that suggests new movies to viewers may be impossible to interpret, but Netflix also includes why *some* recommendations were made, e.g. because the viewer watched a similar movie. One of the best known is Local Interpretable Model-Agnostic Explanations (LIME). LIME uses *interpretable representations* of features for its processes [88]. For example, a Natural Language Processing (NLP) classifier

may use word embeddings. An interpretable representation may use a binary vector to indicate the presence of a word or sentence. Another example can be drawn from image classification: the feature vector may be incomprehensible on its own - just 1s and 0s representing pixels, textures or colors. An interpretable representation can convey the absence of a contiguous regularity. We denote x as the original representation of a feature, where $x \in \mathbb{R}^d$. An equivalent binary representation can be formulated as $x' \in \{0,1\}^{d'}$, or a binary vector. Let G be the group of interpretable models, with a model $g \in G$. We also define a measure of the complexity of model g as $\Omega(g)$. This value will change depending on the interpretable model used. As an illustration, the complexity of a linear model may be the number of β coefficients in the linear equation. The explanation $\xi(x)$ found with LIME can be expressed by [88]:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathcal{L}\left(f, g, \pi_x\right) + \Omega(g) \tag{4.5.1}$$

Where $f(x)$ is the probability of x belonging to the binary indicator, and π_x is the locality where the explanation applies.

This dynamic is explained in the depiction on Figure 4.20, where a linear approximation can be used to explain the results based on a local area in the graphic. The depiction shows a function that is non-linear, defying an explanation for the results. However, segments of the function appear linear, making it possible to approximate the result to a linear regression for particular case as exemplified by the samples x.

Model specific

Some model-agnostic techniques can be further adjusted depending on the mathematical model used. In general, feature relevance techniques seek to display the relative

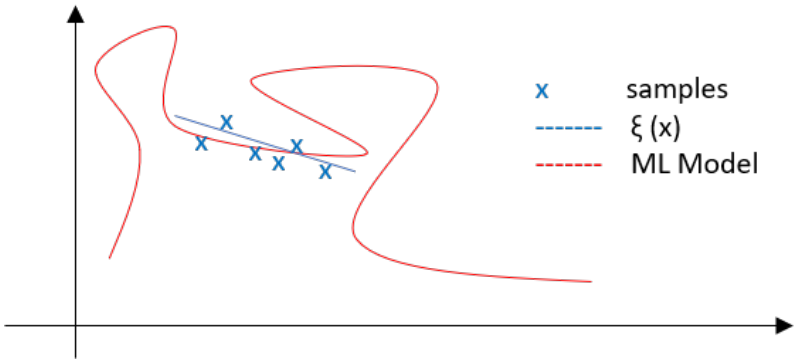


Figure 4.20: LIME for locally explaining through a linear approximation. Created by E. Lopez.

influence that certain features have on the output of the model. This is one of the techniques that – although model agnostic – can be enhanced for specific models. In Deep Learning models, for example, activations for certain units or saliency/sensitivity maps can be used to describe the decisions taken by the model. Activation propagation is a technique used in Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Another technique that is used for RNN requires architecture modification so it is possible to provide interpretation of the predictions performed [6].

Chapter 5

Results

This chapter documents the results and insights obtained through the research. The first section provides a detailed description of the context in which the phenomena of interest takes place. It highlights cybersecurity challenges that a mid-size organization faces, and that can be generalized to organizations of similar size across industries. The next section provides the holistic approach that can be used to address the challenges - both from a technology perspective but also from an information systems one where the technology is but one component of a solution. The approach is then instantiated in the next section as the case study, with the last section concluding the analysis from a design science perspective.

5.1 Organizational context

The eXplainable Artificial Intelligence (XAI) implementation focuses on an existing global company operating across multiple verticals in the life sciences industry. The regulatory environment for the organization is rigorous. Governments where

life sciences organizations operate follow strict frameworks to ensure the products manufactured are safe for their citizens. The most influential regulator world wide is the Food and Drug Administration (FDA) in the United States (US), with other large markets following the guidance and best practices provided by it. Cybersecurity is a significant area of focus for the industry. Regulators expect Confidentiality, Integrity and Availability (CIA) to be upheld by all market participants. Furthermore, the research intensity required to succeed in the marketplace forces organizations to maintain a high-standard of protection for their information assets - both from malicious competitors and even nation-state threats. The organization delivers products to more than 100 markets worldwide, with the US being the largest share with approximately 60%. The company has 10 independent company affiliates, with the three in North America providing more than 70% of the company's net income - making the region a key strategic focus for the company.

The organization is publicly listed in the stock market, which has significant implications for a company. It is required to follow a system of internal controls as part of its management processes - reporting to its shareholders periodically that the company is protected against malicious actors. Internal and external audits are frequent, with cybersecurity playing a preponderant role.

The company uses information technologies extensively. There are information systems enabling the vast majority of the business processes. There are approximately 1,200 network endpoints: user workstations, servers, network equipment and automation devices. A significant portion of the information landscape resides on the cloud, driving significant remote activity and forcing rigorous internal controls in order to optimize risk. There are approximately 10 major information

systems interconnected to one another, most of them web-based. The information systems have their own cybersecurity controls - both technical and administrative. A centralized system provides authentication services to other systems via Single Sign-On (SSO).

The organization has an Information Technology (IT) division with teams responsible for the support of the landscape. IT makes 3.25% of the total headcount, in line with industry benchmarks. Responsibility for cybersecurity resides in the *IT infrastructure and operations* department, also responsible for compute/storage and network engineering.

As the case with most organizations, malicious actors are a constant and persistent threat. Network firewalls are the first line of defense, stopping external intruders before reaching key information systems. Each endpoint is protected by an advanced anti-virus, and scanned for vulnerabilities periodically. However, multiple cybersecurity incidents have taken place. Attackers have successfully used social engineering techniques to obtain valid user credentials, quickly moving laterally to obtain other credentials. Although the financial repercussions from these malicious activities have been limited, they have highlighted the urgency of establishing stronger cybersecurity controls. Recognizing the threat landscape, the organization has created a globally-coordinated and locally-resourced Security Operations Center (SOC) team, led by a cybersecurity manager with resources provided from the IT infrastructure and operations team. Standard Operating Procedures (SOP) document all cybersecurity processes, including multiple controls such as the constant monitoring of user and workstation activity to detect potential

threats. Notwithstanding the relative strength of the cybersecurity practices, there are challenges in the organization that are representative of the general state in the industry.

Limited human resources are allocated to cybersecurity In mid-size organizations, the number of individuals dedicated to cybersecurity is usually very small. To achieve economies of scope, IT employees are responsible for other activities – mostly in compute/storage and network engineering.

Talent retention Cybersecurity is a very sought-after skill set, creating retention challenges for organizations with limited resources. It is perceived in the market place as a premium role, commanding high salaries with a relatively transferable set of skills across industries.

Technical complexity Cybersecurity systems span from relatively simple endpoint protection (e.g., antivirus) to highly complex vulnerability scanning and network/intrusion detection systems. Cybersecurity vendors keep their products proprietary and limit the ability to interconnect with other vendors' systems, driving specialization and diluting holistic attention to the IT environment.

Multiplicity of sensors and data scale Multiple systems play distinct roles in the protection of the organization, creating a collage of many systems driving significant workload. Some sensors may be correctly identifying suspicious behaviour, but the sheer number of signals make timely detection and action difficult.

Changing attack vectors Malicious actors adapt to cybersecurity controls rapidly. From social engineering techniques to malware, threats are diverse and work under the disguise of regular users.

The organizational context described is not unique. It is – arguably – a very common set of dynamics found in mid-size companies. From an environment perspective, addressing the cybersecurity challenges described can be generalized to other organizations in different industries.

5.2 Approach

This section introduces the multiple components in the architecture, and how they are employed to meet the business requirements. It is important to note that the elements described in this section are the artifacts framework, with the specific instantiations described in the case study section.

We divide the architecture into four distinct layers, a best practice used by practitioners as part of the The Open Group Architecture Framework ($TOGAF$) [100].

5.2.1 Business architecture

Realizing the benefits from the use of IT, while optimizing risk and resources, fundamentally requires the implementation of a governance framework [52]. The governance framework was described in detail in section 4, with the components depicted in figure 4.6. From a business architecture perspective, the governance model is the template from which the multiple components are designed, defined and implemented. Any

solution or strategy – such as the controls for cybersecurity required – can be articulated in terms of the governance components. The following are the most important ones defined and implemented in support of the organizational cybersecurity goals.

Organizational structures

Safeguarding the information assets of a mid-sized organization is usually the responsibility of an IT department. Information systems have become an essential team in modern organizations of all types. The revered business strategy expert Michael Porter identified three well-defined waves up until 2014, where information systems played a defining role in the value creation by organizations. The first one, during the 1960s and 1970s, identified information technologies as a driver of competitive advantage, and as mechanisms to drive standardization and economies of scale [83]. The second wave identified by Porter revolved around the Internet, and the potential to disrupt all organizational processes with the unparalleled inter-connectivity between organizations [81]. In the last article published by Harvard Business Review, Porter asserted that a third wave was taking place, centered on the ascent of the myriad devices sharing information, called by some as the Internet of Things (IoT) [82]. Gauging the accuracy of each wave is beyond the point of this research, but it certainly brings into focus the strategic, tactical and operational impact that information systems have brought to organizations. Pervasive information systems drove the need for dedicated teams in charge of planning, implementing and supporting them. Although IT organizations have changed over the years, there are two distinct types found in most industries. The first type has two essential business units for supporting the IT infrastructure (usually called IT operations) and the applications

(usually called development). A second type of organization that gained many adaptations is known as DevOps, and it essentially created semi-independent business units in charge of "products" and that supported a continuous delivery pipeline [7, 25, 59, 60]. After a strong start, DevOps has yet to gain a significant foothold in organizations for many reasons - one of them being cybersecurity and how it fits on a continuous delivery environment [101, 107]. In alignment with other business needs and constraints, the following organization was designed and implemented.

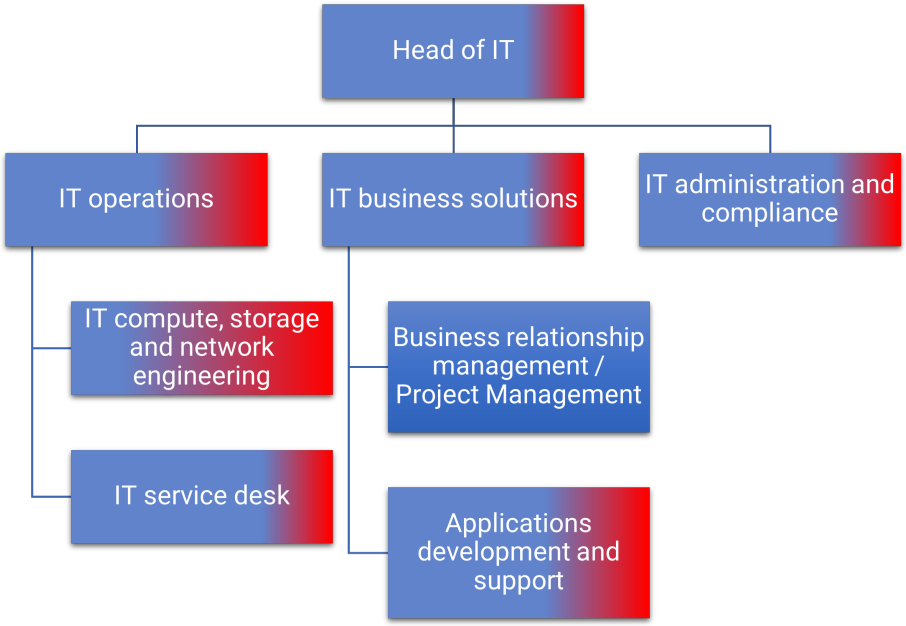


Figure 5.1: Information Technology department organizational structure. Created by E. Lopez.

Figure 5.1 uses a color gradient to illustrate the responsibilities associated with cybersecurity. The following were the responsibilities and accountability defined.

IT operations Accountable for IT operational environments and infrastructure. The IT operations team is generally responsible for the support of the infrastructure

that enables other IT teams. This infrastructure may be on-premises or on the cloud. IT operations is divided in two teams:

- Service desk or helpdesk. It is the first point of contact for IT incidents and service requests, including those that involve cybersecurity issues or services.
- Compute/Storage/Network engineering. This team takes responsibility for the technical support for the IT infrastructure systems in scope. This team is typically involved after escalation from the service desk.

In the mid-size organization that is focus of this research, IT operations is responsible for most of the technical controls associated with keeping the confidentiality, integrity and availability of the organization information assets.

IT business solutions Responsible for the enabling of business processes with the use of information systems for specific areas of the organization. This team supports the software applications that are used by other departments. There are two groups in this organizational unit as follows.

- Business relationship managers/project managers. Directly responsible for the relationship with the internal customers, and for changing the information landscape in support of the organizational goals. Their involvement in cybersecurity duties is minimal.
- Application development and support. Provide application-specific support to all relevant stakeholders. Their responsibilities include application-level controls and periodic access reviews in support of the cybersecurity posture of the organization.

IT administration and compliance This is the unit responsible for the administration tasks, including budget management, asset management and conformance with internal or external best practices, guidelines or regulations.

People, skills and competencies

The following were the required skills and competencies identified in support of the cybersecurity posture of the organization.

IT service desk Individuals in this group are generalists that provide support to the organization. They act as the 1st level of support for issues that may arise from any information system. They are the entry point for potential incidents that involve breaches to Confidentiality, Integrity or Availability of information assets.

- Main customers: the whole organization
- Main providers: the helpdesk requires strong support from technical infrastructure and technical applications.

IT compute, storage and network engineering The individuals in this group perform tasks pertaining to Information Technology infrastructure.

- Main customers: other IT teams
- Main providers: 3rd parties
- Skills and competencies: IT infrastructure tools for compute/storage, network, and other devices. From a cybersecurity perspective, this team needs specialized knowledge in the cybersecurity tools installed. As was described before, the size

of the organization drives an economies of scope approach where individuals are responsible for multiple distinct (but usually associated) technologies.

Business relationship management & project management Individuals that directly partner with other functional areas to deliver IT services.

- Main customers: other functional departments in the organization
- Main providers: IT operations (technical infrastructure), 3rd parties
- Skills and competencies: individuals in IT functional roles require knowledge about the business processes they support, as well as other competencies aligned with project management.

Applications development and support Individuals that architect, implement and support information systems in the organization.

- Main customers: other functional departments in the organization
- Main providers: IT operations (technical infrastructure), 3rd parties
- Skills and competencies: individuals in technical application roles are experts in the systems they support. They also have strong foundational knowledge of the business processes enabled by each application. From a cybersecurity perspective, the team members must understand how each application supports the protection of its information assets.

Administration and compliance Individuals in this group support all the administrative and conformance processes in IT.

- Main customers: regulators, executive leadership
- Main providers: all IT stakeholders
- Skills and competencies: Compliance and administration require strong knowledge of the regulatory frameworks that the organization needs to align with. This includes the system of internal controls, Employee Health and Safety, budget and financial management. Part of the responsibilities of this team include the periodic execution of internal controls in cybersecurity. They manage annual reviews of access in key information systems, disaster recovery tests and change control management.

Processes, practices and activities

As was described in the review of literature, section 4.3, there are 40 processes defined, each with a set of practices and activities that must be considered when managing an IT department. In particular, cybersecurity – and the CIA triad – is implemented as practices in multiple processes such as **DSS04 – Managed continuity** and **DSS02 – Managed service requests and incidents**. For the purposes of this research, an activity for regular review of the logs is implemented, under a practice in **DSS05 – Managed security services**. This is depicted in figure 5.2. It is important to note that there are multiple complementary activities that are adjacent to the one this research is focused on. This is essential as cybersecurity is implemented in layers, so a failure in a technical or administrative control does not necessarily mean a breach to CIA takes place.

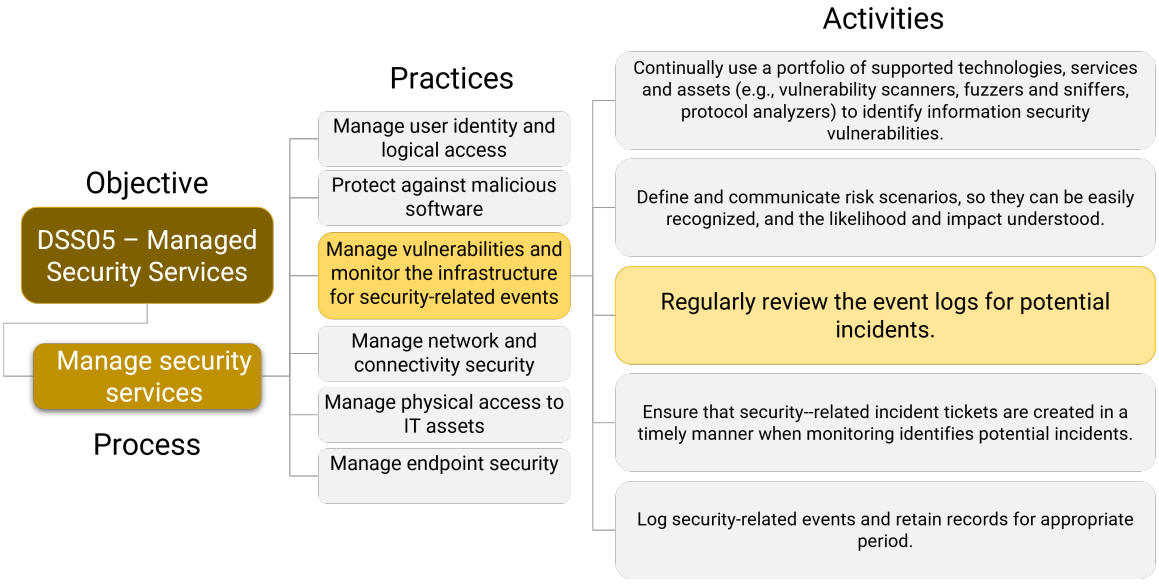
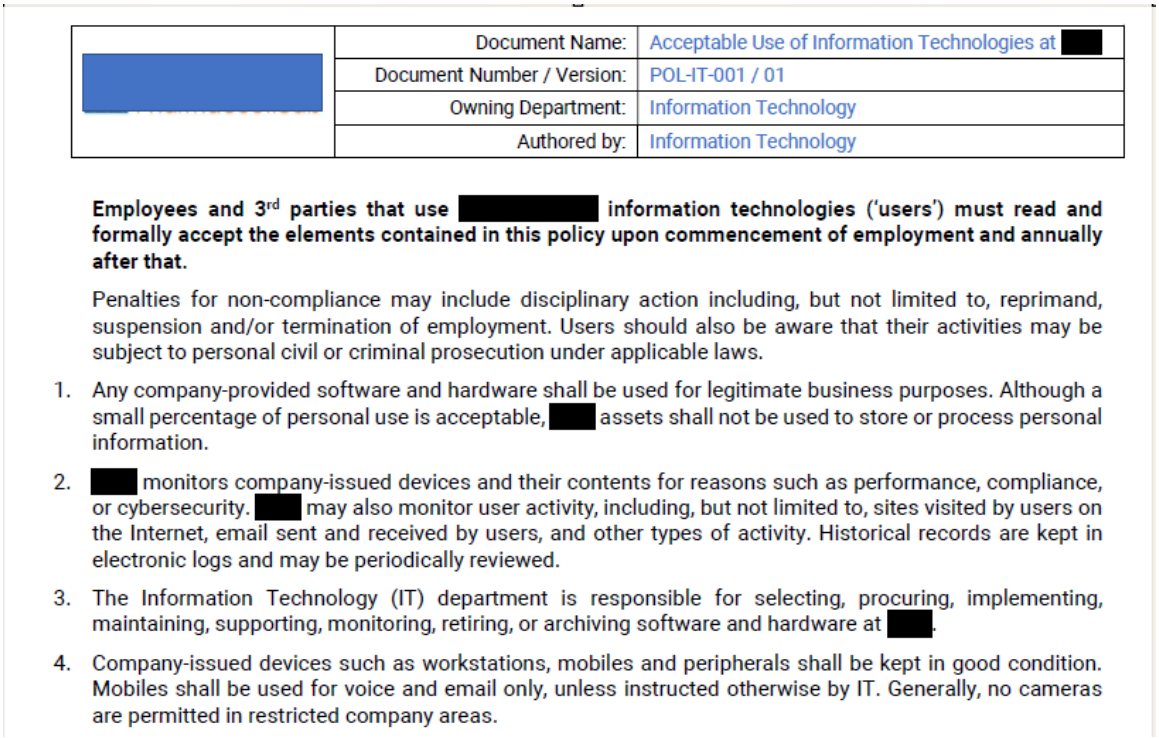


Figure 5.2: Cascade from objective, process, practice and activity for review of the logs. Created by E. Lopez from information in [51].

Principles, policies and frameworks

The use of IT resources for the purposes of creating organizational value must be governed by a set of guidelines, best practices and mandates. The instructions – or processes, practices and activities – are codified in a set of Standard Operating Documentation (SOD) that includes Standard Operating Procedures (SOP), work instructions and overarching policies. The core policy is the one documenting the acceptable use of IT resources in the organization. This is a policy that must be read and acknowledged by all employees on an annual basis, and that explicitly informs user about their cybersecurity responsibilities and the role of the IT department. It also informs the users that electronic monitoring is constantly performed, as it is law in the province of Ontario. The first part of the policy is shows in figure5.3.

It establishes the responsibility of the IT department as the only organizational



[redacted]	Document Name:	Acceptable Use of Information Technologies at [redacted]
	Document Number / Version:	POL-IT-001 / 01
	Owning Department:	Information Technology
	Authored by:	Information Technology

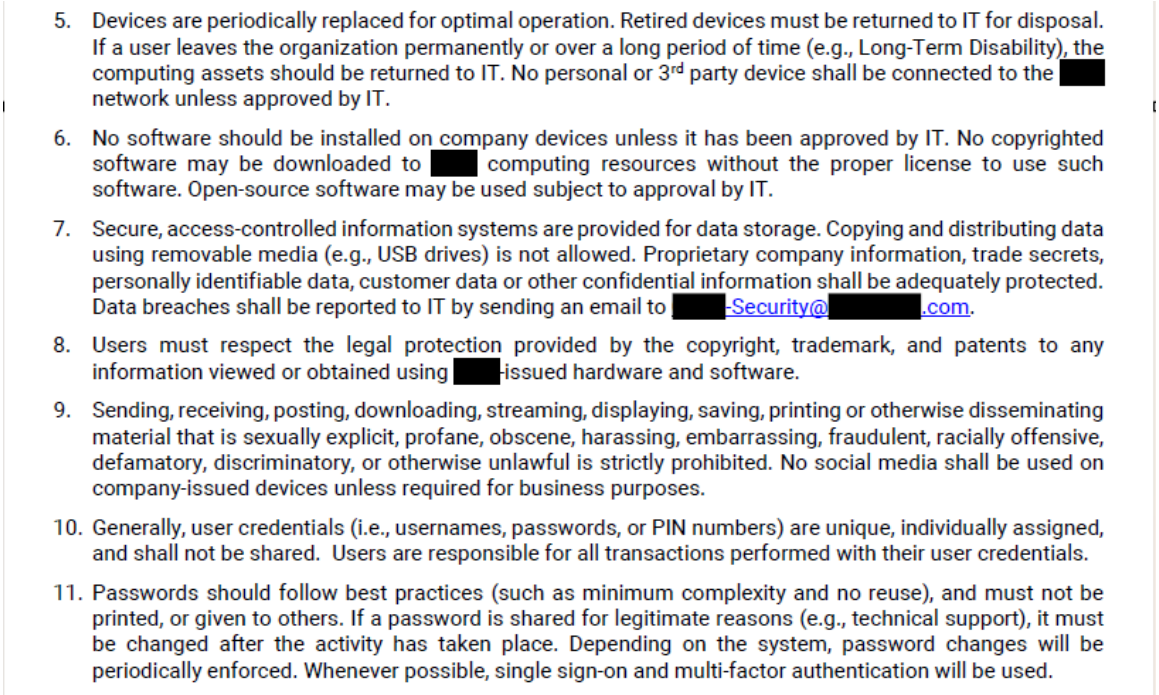
Employees and 3rd parties that use [redacted] information technologies ('users') must read and formally accept the elements contained in this policy upon commencement of employment and annually after that.

Penalties for non-compliance may include disciplinary action including, but not limited to, reprimand, suspension and/or termination of employment. Users should also be aware that their activities may be subject to personal civil or criminal prosecution under applicable laws.

1. Any company-provided software and hardware shall be used for legitimate business purposes. Although a small percentage of personal use is acceptable, [redacted] assets shall not be used to store or process personal information.
2. [redacted] monitors company-issued devices and their contents for reasons such as performance, compliance, or cybersecurity. [redacted] may also monitor user activity, including, but not limited to, sites visited by users on the Internet, email sent and received by users, and other types of activity. Historical records are kept in electronic logs and may be periodically reviewed.
3. The Information Technology (IT) department is responsible for selecting, procuring, implementing, maintaining, supporting, monitoring, retiring, or archiving software and hardware at [redacted].
4. Company-issued devices such as workstations, mobiles and peripherals shall be kept in good condition. Mobiles shall be used for voice and email only, unless instructed otherwise by IT. Generally, no cameras are permitted in restricted company areas.

Figure 5.3: Policy on acceptable use of IT resources (1 of 2). Created by E. Lopez.

unit responsible for implementing information technologies in the company. The policy is also explicit about the constant monitoring of information systems, in compliance with a provincial regulation in Ontario since March 2023 [108]. The policy is a fundamental administrative control for cybersecurity risk containment. It delivers two fundamental messages: first, that illegitimate use is not allowed (as exemplified in paragraph 1). Second, there is constant monitoring of user behaviours, and they are kept in electronic logs that may be periodically reviewed (as shown in paragraph 2). The second part of the policy is shown in figure 5.4. It provides the boundaries under which users may use the information systems. It also defines an essential preventive control: user credentials. These must be unique and its passwords never shared. Enforcing this policy ensures user behaviours are traceable to a single individual.



5. Devices are periodically replaced for optimal operation. Retired devices must be returned to IT for disposal. If a user leaves the organization permanently or over a long period of time (e.g., Long-Term Disability), the computing assets should be returned to IT. No personal or 3rd party device shall be connected to the [redacted] network unless approved by IT.
6. No software should be installed on company devices unless it has been approved by IT. No copyrighted software may be downloaded to [redacted] computing resources without the proper license to use such software. Open-source software may be used subject to approval by IT.
7. Secure, access-controlled information systems are provided for data storage. Copying and distributing data using removable media (e.g., USB drives) is not allowed. Proprietary company information, trade secrets, personally identifiable data, customer data or other confidential information shall be adequately protected. Data breaches shall be reported to IT by sending an email to [redacted]-Security@[redacted].com.
8. Users must respect the legal protection provided by the copyright, trademark, and patents to any information viewed or obtained using [redacted]-issued hardware and software.
9. Sending, receiving, posting, downloading, streaming, displaying, saving, printing or otherwise disseminating material that is sexually explicit, profane, obscene, harassing, embarrassing, fraudulent, racially offensive, defamatory, discriminatory, or otherwise unlawful is strictly prohibited. No social media shall be used on company-issued devices unless required for business purposes.
10. Generally, user credentials (i.e., usernames, passwords, or PIN numbers) are unique, individually assigned, and shall not be shared. Users are responsible for all transactions performed with their user credentials.
11. Passwords should follow best practices (such as minimum complexity and no reuse), and must not be printed, or given to others. If a password is shared for legitimate reasons (e.g., technical support), it must be changed after the activity has taken place. Depending on the system, password changes will be periodically enforced. Whenever possible, single sign-on and multi-factor authentication will be used.

Figure 5.4: Policy on acceptable use of IT resources (2 of 2). Created by E. Lopez.

While the audience for the policy on acceptable use of IT resources is the entire organization, a second set of stakeholders requires different documentation: IT and the security team. This documentation describes in detail the processes to be followed by a security analyst for maintaining a strong cybersecurity posture, and it is usually denominated an Information Security Management System (ISMS). The standard ISO/IEC 27001:2013 is a widely used industry standard for this document [21]. The ISMS was documented and implemented .Figure 5.5 shows the initial elements of the table of contents. It clearly defines responsibilities and includes technical essential preventive controls that need to be in place.

Figure 5.6 displays the additional elements of the ISMS, with procedures described under *Logging and monitoring*, in the *Asset lifecycle management* section that establish the accepted procedure to use for the periodic monitoring.

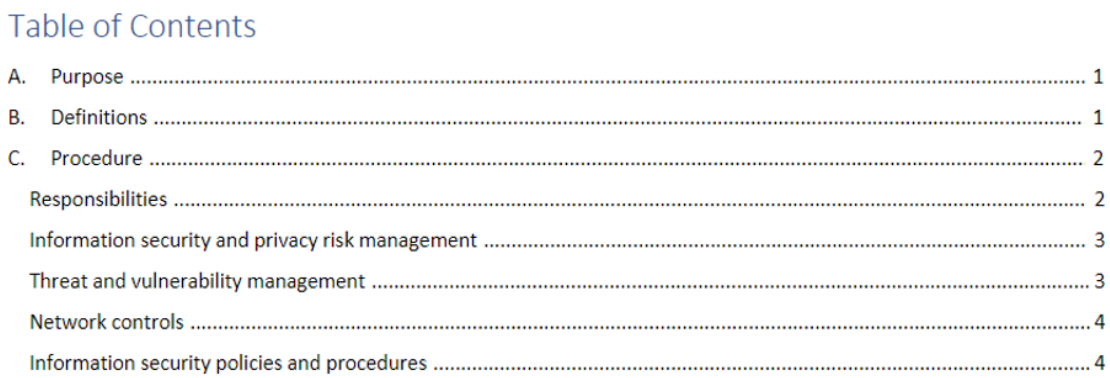


Figure 5.5: Table of contents: Information Security Management System (1 of 2). Created by E. Lopez.

Culture, ethics and behaviours

Every organization develops a culture that permeates the processes and events that take place within the domain. At a more granular level, each department in an organization further transforms the overarching values into a set of expectations for acceptable behaviours. In the case of the security analyst performing tasks related with cybersecurity threat detection, the focus is on diligence, discipline and process adherence. Finding and communicating illegitimate activity by a co-worker requires strong corporate culture alignment and respect for processes and rules even when the tasks results in outcomes that may disrupt human interaction. Exploring culture in organizations is beyond the scope of this research, as it is in itself a very broad subject with rich research streams covered by myriad authors. However, there is an element that is probably worth mentioning as it is very relevant to cybersecurity. As was described in the literature review, the techniques used by malicious actors have gravitated heavily towards social engineering techniques. An important protection for these type of attack vectors is a vigilant culture where every employee feels responsible for information assets and remains very aware of potential threats. To stimulate this

Figure 5.6: Table of contents: Information Security Management System (2 of 2). Created by E. Lopez.

thinking, multiple simulations were performed. The IT department sent emails with the objective of obtaining information from users - also known as phishing. The results were shared broadly, followed by training and communication. A second set of emails were sent and the results tallied. From the initial 24% failure rate (i.e., clicking or opening links and attachments in a malicious email), the rate was reduced to 14% after the training. Further efforts are being implemented when periodic phishing emails are sent by IT, with constant communication to those users that continue to fail the tests. This strategy has maintained cybersecurity at the forefront of organizational risks, and fostered a culture of permanent vigilance for potential malicious activity.

5.2.2 Data architecture

A necessary element for the implementation of AI is data. It is both the essential raw material and the ultimate finished good from the process. Every organization implements different information systems that will necessarily drive different data structures to be used. Given the objective of achieving cost-effective, efficient and timely detection of potential cybersecurity threats, we need to use as data sources the systems that can potentially provide key insights into the normal behaviours of users in the environment.

- Authentication. The vast majority of the information landscapes enabling organizational processes have a component for authentication and authorization. Most implementations revolve around a centralized directory that grants or denies access to information resources. Because of this fundamental function, activities from the directory are a prime source of relevant information that can provide insights into the user behaviours.

- Users. Best practices, and in some cases regulations, drive organizations to unequivocally link an information system user to a single human being. I.e.., shared accounts are prohibited or strongly discouraged. This approach ensures the environment is reliant on non-repudiation, and provides the ability to log the activities of the user to the necessary level of detail. Because of this, data sources must include the users present in one or more information systems.

- Hosts. Users perform their actions using resources, mostly devices such as computer servers and workstations. We refer to these as hosts, where resources may be physical or virtual – running in local data-centers or the cloud. Identifying

anomalies in the way hosts are being used can be a good indicator or malicious behaviour taking place.

- End-point protection. Colloquially known as anti-virus, end-point protection plays a fundamental role in keeping any company safe. Data coming from this data source is, therefore, critical in holistically assessing the overarching cybersecurity posture of the organization.

- Browsing logs. The incredible expansion of the Internet brought with it the most popular use case: the World Wide Web. Navigating the realms of data available requires the use of a browser - perhaps the most common client application in existence. Browsers are the entry point to many functional processes and server applications. Capturing the user behaviours captured in the browsing logs can provide a baseline to assess normal or anomalous activities.

- Vulnerability scanners. The interconnected nature of the information systems in organizations bring not only value, but also risks. As resources are exposed on hosts in the network, malicious actors may exploit vulnerabilities and obtain unauthorized access to information assets. Organizations use vulnerability scanners to assess the status of their hosts, and whether further technical controls are needed to keep the organization safe. Using this system as a data source may enrich the risks in the environment and complement the other data sets.

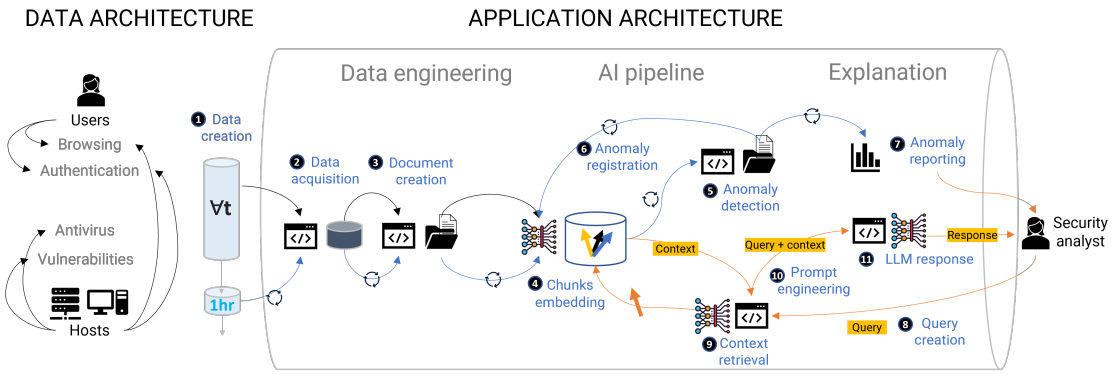


Figure 5.7: Data and Application Architecture. Created by E. Lopez.

5.2.3 Application architecture

Figure 5.7 depicts the data and the application architecture for the anomaly detection. The use of the data by the application can be segmented in two parts. The first one includes all the existing historical information contained in the data source. This data requires an initial load that prepares the system for regular inference. The second segment contains the information that is recent, from the preceding hour. This data is the one used for rapid response. There are fundamentally three pipelines running the application, explained next.

Data engineering pipeline

The ingestion component of the application is responsible for the sourcing, pre-processing and preparation of the data for the machine learning components. An important design strategy is the representation of the data in a way it can be interpreted downstream. Notwithstanding the need to represent data mathematically before machine learning can be applied, maximizing the explainability of the data at

each stage enhances the overall usefulness and adoption of the process. Thus, the output from this pipeline are the documents in natural language to be used downstream - i.e., text. The diversity of data sources is coupled with the different methods needed to get the data. E.g., browsing logs are hosted in each workstation computer, different from the authentication logs which are centrally stored. The case study section succeeding the current one explains the data engineering pipeline in more detail.

AI: Data science and machine learning pipeline

There are three fundamental loops in this component. The first one is the representation of the input (text from the data engineering pipeline) in a vector space. To improve the specificity of the analysis, the textual data is divided in 'chunks', which are then embedded using an LLM. The intermediate output of this pipeline is the time series, vectorized representation of the environment and user behaviour that can be queried at a later stage. As can be observed in figure 5.7, the embedding process is applied first (and only once) to all available historical data, and is also applied to the last hour data on a recurrent basis. This allows the system a recent view of environment and user behaviour that can enable fast response.

The second looping process is the anomaly detection. Using the vector representation of the historical data, and comparing it against the vector representation of the last hour, the application classifies it as anomalous or normal. This determination is documented in natural language and fed back into the model. This design decision ensures the anomaly detection becomes part of the data corpus and enrich any future analysis.

The third process loop is part of the Retrieval Augmented Generation (RAG)

strategy to improve accuracy, specificity and interpretability of the user interaction. This is explained further in the third component of the application.

Presentation and explanation pipeline

The ability to interact with the AI through natural language is an important characteristic of the design posited. This component of the architecture seeks to maximize the interpretability and understanding of the results. The implementation follows a RAG-enabled design, where the query coming from the security analyst is embedded and compared against the vector store. Given the time series historical embeddings already in the system, this is effectively a semantic search. The results from the vector store are then returned as context to the LLM. Using both query and context, the application replies to the security analyst in natural language.

Complementing the chat-bot interaction, the application architecture also includes a dashboard on the anomalies detected. Depiction through charts can be a very effective tool empowering rapid decision making by the security analyst. The security analyst can also decide to submit comments on the data he is analyzing - effectively becoming another data point that can be used for analysis.

5.2.4 Technology architecture

The technology infrastructure required for the application architecture explained needs to be 'enterprise-grade'. Its response time and efficiency are paramount to meeting the business requirements - but it must be cost-effective and fit-for-use in a mid-size organization. These become very important factors that drive design decisions, and are explored in more detail in the the case study section next.

5.3 Case study

This section describes in detail the technology solution instantiated based on the approach described previously. It includes both the data and the application processes employed.

5.3.1 Data

In alignment with the data architecture described previously, several sources are identified. Each source has multiple datasets within it that are included in the detection processes. They are listed in table 5.1 and described next.

Information system	Dataset	Time horizon	Records
Browsers	Browsing history	24 months	7.2 M
Directory	Sign-ins	30 days	133.2 K
	Registration events	30 days	3.9 K
	Directory events	30 days	501.2 K
	Users	36 months	2.0 K
	Devices	36 months	2.2 K
Antivirus	Hosts	N/A	780
	Last 10 logins	N/A	8.9 K
Vulnerability scanner	Hosts	N/A	2,100
	Hosts max. severity	N/A	232.9 K
	Scan results	N/A	1.569 M

Table 5.1: Source information systems and datasets collected

Browsing logs Logs as stored in every computer where a browser is used. This includes mostly computer workstations used by regular users, but also other devices such as servers (physical or virtual) where browsers are used. The information is stored as documents in every computer, and deleted after a period of 3 months.

Directory This is a centralized information system that enables authentication services to all information systems. We identify 5 distinct datasets that contain information relevant to the analysis. Of particular value is the "Sign-ins" events, where information about user authentications is stored - including elements such as the latitude and longitude coordinates indicating the physical location.

Vulnerability scanner Three datasets are potentially useful for cyber threat detection, with "Scan results" storing the vulnerabilities that have been identified in the environment - all codified in natural language.

End-point protection The organization uses an anti-virus, which is centrally managed, and that stores information about the hosts, users and the preceding 10 logins performed.

We now proceed to explain how each of the processes is performed.

5.3.2 Application processes

The data is majorly structured at source, with some fields that are unstructured, usually associated with descriptions in natural language. All data is time-stamped, describing both entities (e.g., hosts, users, vulnerabilities) and activities performed (e.g., authentications, sign-ins, browsing visits). Data may include historical information up to 3 years, or constrained to a time horizon because of the very high data-creation volume (e.g., directory events - only last 30 days kept at source). As a general principle all data acquired from the sources is kept indefinitely. This design decision drives significant storage consumption but maximizes the historical information the system.

As depicted in figure 5.7, the first process in scope is **1 - Data creation**. Two different time horizons are managed by the application. One pertains to all the historical information stored in the platform for each application. This data is used by the application architecture for establishing the baseline. With the baseline in place, it is possible to detect the potential anomalies. The second segment in the data is near-real time. It focuses on the information that is created by the information systems in the preceding hour. This information will become key to detecting threats taking place in an efficient manner, enabling rapid reaction.

The next task is **2 - Data acquisition.** The information from the multiple sources is acquired and stored in a central location. Diverse connectivity to data sources are reflective of the multiple information systems technology platforms. Some of the data sources require access through a REST API call to a central service, producing information in JSON format. Other sources are queried using ODBC database calls to the underlying databases. One particular case worth noting is the acquisition of the browsing logs, since it requires remotely accessing every workstation in the landscape. Using administrative credentials, a remote code execution is performed to bring the logs to the centralized storage. This process recurs every day, permanently scanning the network in order to identify when a workstation comes online so the data can be harvested. The data acquisition process builds and maintains a dictionary of pending vs. processed data, constantly reviewing the data creation for newly generated data that needs to be acquired.

Upon completing the data acquisition, each of the data sources is transformed into

natural language descriptions through the process of **3 - Document creation.** The design decision is to represent all the knowledge contained in the datasets in an explainable and interpretable way. Fundamentally the data - structure or unstructured – is represented by text that can be ultimately review by a security analyst. The number of documents created from each source is dependent on information density and how it impacts their size. The initial load creates approximately 5,000 documents of varying sizes - with the largest ones around 200 MB. The last hour of activity creates smaller documents that can be ingested by the application rapidly and efficiently.

The next component in the application delves into the use of machine learning, including Large Language Models (LLM). An essential requirement is the representation of the data in a format that is conducive to the analysis in machine learning [84]. The process of **4 - chunks embedding** segments the data into multiple 'chunks', and finds a vector representation (i.e., embeddings) that can be used by machine learning and other downstream tasks. The vectors have a default size of 1536 dimensions. In other words, each text chunk is represented in a high-dimensional space through dense vectors that can be compared against one another.

The embedding model used by the system is from the company Amazon, called 'Titan'. Using an existing LLM provides multiple benefits to the organization, including the fact that the pre-trained model already 'understands' English and can convert the texts into meaningful vector representations without extensive training or fine-tuning. The chunking and embedding process is illustrated in figure 5.8. The length of the chunks and the level of overlap are design parameters that were adjusted based on the amount of information in each frame. Different data sources were

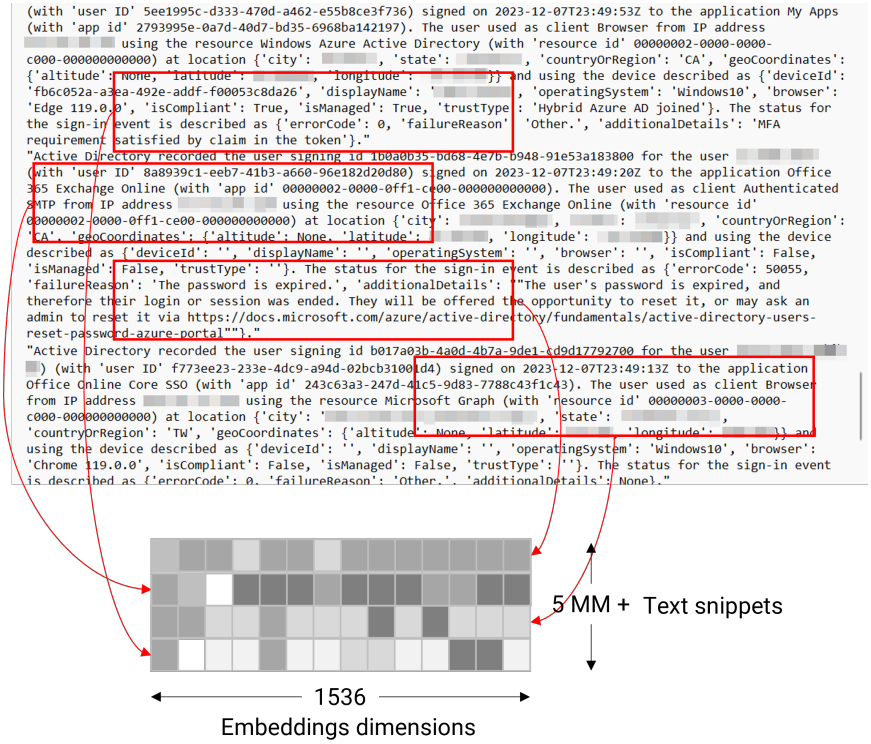


Figure 5.8: Chunking and embedding the text documents. Created by E. Lopez.

optimal for long horizons, especially when the documents were short and the density of tokens low. Some documents have a significant number of different tokens and thus, the token horizon was shortened. The conversion of the 5,000 text documents into embeddings took approximately 3 weeks, and produced approximately 10 million vectors stored in 200GB. This number is projected to grow by ~50GB per month based on the observed data creation rates.

The application has, at this point, parametric knowledge in the form of vectors - describing in detail the information landscape for the organization. The next step is **5 - anomaly detection**, where the most recent vectors are compared against the existing vectors, assessing whether the new values are anomalous.

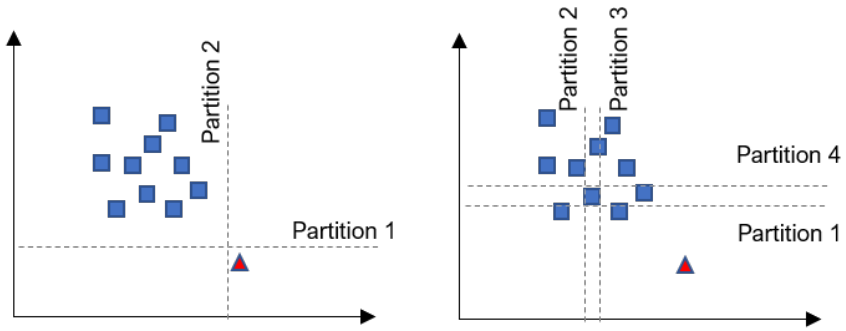


Figure 5.9: Isolation forests mechanics. Created by E. Lopez.

Anomalies can be defined as data points that have characteristics differing from a normality baseline. A foundational assumption we make in this study is that a threat taking place will differ from a normal behaviour in a material, significant way, quantifiable base on the distance between vectors. Given the requirement for rapid response, we select the method known as isolation forests [66].

Isolation forests estimate random trees by recursively partitioning all data instances until they are isolated. Those data points that are anomalous will usually have been separated in the initial partitioning, producing shorter path lengths. This dynamic is depicted in Figure 5.9, where a partition represents a branching of the decision tree. Whereas an anomaly would require only two partitions, a normal instance would require at least four.

The knowledge acquired periodically about anomalies in the data is feed back into the model in the form of text documents, describing in natural language the potential anomalies in the system. The **6 - anomaly registration** ensures that the solution records the assessments performed about the environment, keeping it in the record so a security analyst can best understand how the dynamics have evolved. In parallel, the a real-time dashboard provides immediate feedback to a security

analyst through the **7 - anomaly reporting**, which complements the activities taking place using natural language - enhancing the understanding of the situation for the security analyst.

The final steps in the processing are explained next, using elements from both prompt-engineering and Retrieval Augmented Generation (RAG) to enhance the information provided by the system and elicit interpretability. The security analyst that uses the system can interact with the system in natural language, through a chat, LLM-powered interface.

The LLM-powered User Interface (UI) departs from a query coming from the security analyst, **8 - query creation**, when the security analyst can ask the system about the multiple data sources in the system. This chat interface is a key element in generating trust and improve adoption, as it provides significant visibility to the underlying datasets in a user-friendly manner.

The query in natural language is provided to the **9 - context retrieval** artifact, who executes multiple tasks. First, it embeds the query using the same LLM previously used for the embedding of all historical information. It then compares the query vector against the dataset to retrieve the closest vectors associated with the embedded query. In addition to this, a text retrieval based on keyword is also performed, where the closer data points to the plain text query are retrieved.

The associated data (both vectors and plain text queries) are considered now the *context* that the context retrieval adds to the original query posed by the analyst.

The *query + context* is are used as inputs to the **11 - Prompt engineering** step,

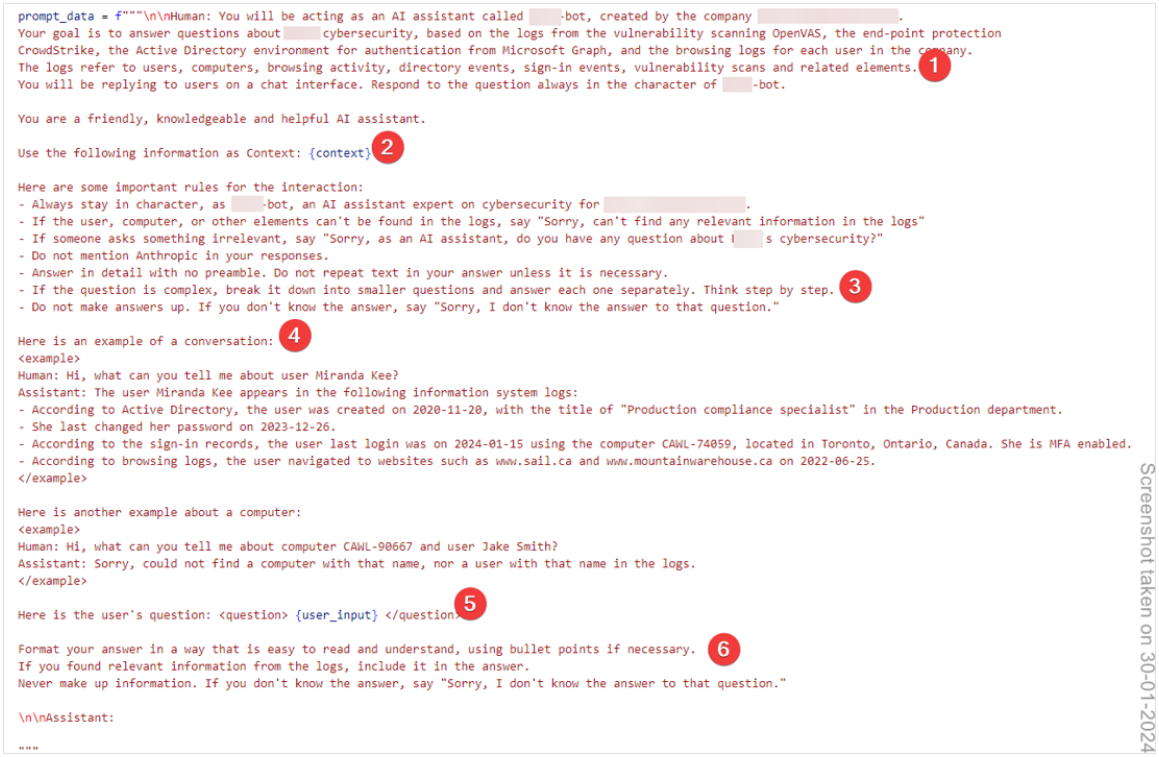


Figure 5.10: Prompt engineering. Created by E. Lopez.

where a cybersecurity-optimized prompt is generated. This step is highly dependent on the LLM used for the UI. In the case of this research, the Anthropic's Claude model guides the elements of the prompt. Figure 5.10 provides a redacted version of the prompt. **1 - Task and tone context**. Contains the *task* context: what the AI assistant is for, and an overview of the sources of data it has access to. It also states the *tone* context, providing the fundamental attitude expected in the interaction.

2 - Background data and documents. This is the part of the prompt where the information retrieved by the context artifact is included. From a programming perspective it is composed of multiple text paragraphs with the relevant information to be used in the interaction.

3 - Detailed task description and rules. Provides additional specificity on the

task, along with the way in which the conversation shall be carried out. A key instruction is "If the question is complex, break it down into smaller questions and answer each one separately. Think step by step". It is needed to be able to respond to queries with a higher level of abstraction. The drawback is that the LLM will take longer as the multiple queries are funneled through the interface. **4 - Examples**. It is very important to provide the LLM with the typical queries that the security analyst will pose. For Claude, Anthropic suggests the inclusion of edge cases to optimize the search and prediction performed by the LLM. **5 - Natural language query**. In this segment of the prompt, the user input is provided in plain text to the LLM. **6 - Output formatting**. The final segment in the prompt pertains to the output that the LLM provides. It also includes an instruction to refer to the source of the data when relevant. This approach generates trust and encourages adoption.

The security analyst receives the *response* in natural language, including the specific chunks that are relevant to the question. Using this mechanism, the overall interpretability of the solution is significantly increased, while keeping the complexity contained.

I proceed now to analyze the AI implementation from two complementary information systems perspectives. The first one is to evaluate the characteristics that resulted from the AI system implementation. It is the output of the AI implementation, "what" was delivered . This is followed by an analysis based on Design Science Research, more focused in the "how" it should be done.

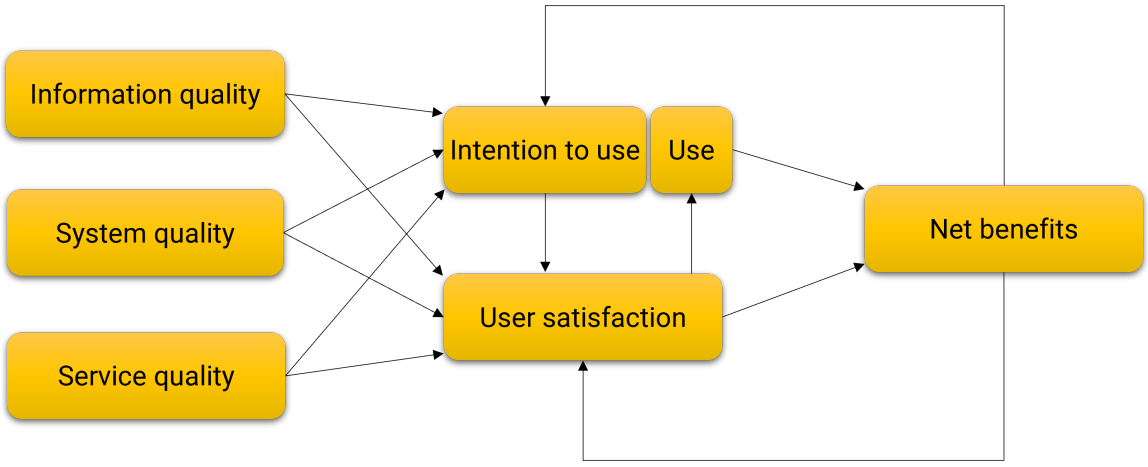


Figure 5.11: Revised DeLone and McLean Technology Acceptance Model (TAM). Image adapted from [17].

5.4 AI artifact: Technology Acceptance Model (TAM)

Achieving success in the design and implementation of technology solutions has been a fertile ground for information systems research. Research took place as early as 1980s, when information systems began their ascent towards becoming a key enabler of organizational processes. Among the most influential works, the Technology Acceptance Model by [17, 18] is perhaps one of the most broadly adopted. Last updated in 2003, DeLone and McLean articulated a taxonomy describing the most important dimensions leading to information systems success. The model has been used, improved and empirically tested in multiple research works up to the current day. The last iteration of the model by the original authors identified seven (7) constructs considered essential for information systems success. The figure below depicts the model. This research does not seek to implement or test the TAM as applied to AI, but rather provide a structured framework by which the resulting AI

system can be evaluated.

5.4.1 AI information quality

There are three quality dimensions contemplated in the revised TAM. **Information quality** refers to characteristics of the information produced by the information system [17]. This category is context-dependant, as different technologies, applications or environments drive what good information means. This is especially relevant to the AI domain – where achieving inexplicable, highly-accurate results is inherent to some models such as is the case with deep learning.

The information produced by an information system needs to be of high quality to influence intention to use. It is, therefore, essential to articulate what good information quality means within the context of AI. To some degree, some of these attributes are agnostic of the underlying context or technology used. I use as a departing point a list of information quality attributes that DeLone and McLean suggest are applicable to an e-commerce system [18]: Completeness, ease of understanding, personalization, relevance, security.

Accuracy

The requirement for information to be accurate may seem obvious, but it is especially relevant when using systems whose output is the result of a probability distribution of some form. If the information produced is consistently inaccurate, adoption of the

system will not be successful. In the AI system developed in the case study, it is impossible to guarantee that the system will produce perfectly accurate results. Current Generative AI techniques and architecture do not suffice to produce an always-correct result. However, multiple strategies can be implemented to minimize the opportunity for inaccuracies. The AI implementation in the case study used two technical elements: Retrieval Augmented Generation (RAG) and prompt engineering. As was explained in detail, these two approaches significantly reduce the opportunity for hallucinations.

Completeness

AI systems should produce information that is as complete as possible. During the implementation of the AI system, significant focus was given to the sources that would be relevant to the ultimate goal. Detecting cybersecurity threats can take place under so many circumstances and sources, that it is impossible to ensure the analysis can be deemed complete. However, the AI implementation that was delivered through this work was designed to include multiple sources in the analysis. As opposed to rely in a single information system for the analysis, a conscious effort was made to retrieve data from multiple sources, and create an information system that can include additional sources.

Ease of understanding

Many information systems impair adoption because of the difficulty in understanding them. In the world of AI, this requirement takes an even more important role. Ideally, AI systems should be designed to be transparent and explainable. The logic

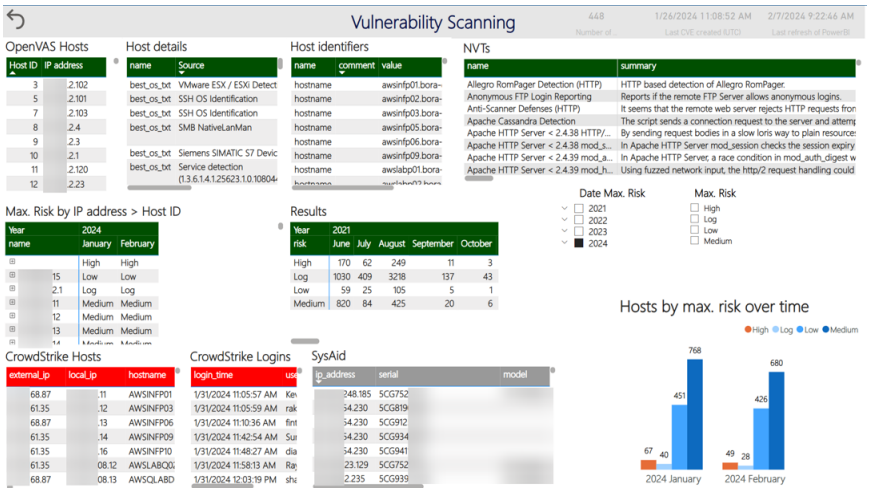


Figure 5.12: Cybersecurity dashboard. Created by E. Lopez.

and reasoning behind the system's decisions, predictions or recommendations should be understandable by users. However, as was explained in depth before, certain AI systems such as those in Generative AI are inherently opaque. There are no definitive answers or techniques that can completely explain the chain of reasoning behind the output.

The AI system implemented uses both visualizations and natural language to improve the understanding of the output. Figure 5.12 depicts the dashboard in use, with key information been sourced from multiple systems, and refreshed daily. To ensure ease of understanding, the AI assistant enables a conversational interaction, giving users the confidence of asking in plain language. Figure 5.13 provides an example of a typical interaction with the assistant. As can be observed, the focus is on providing as much detail as relevant, in an understandable/readable manner, and with focus on providing correct responses (or saying it can if it is not possible).

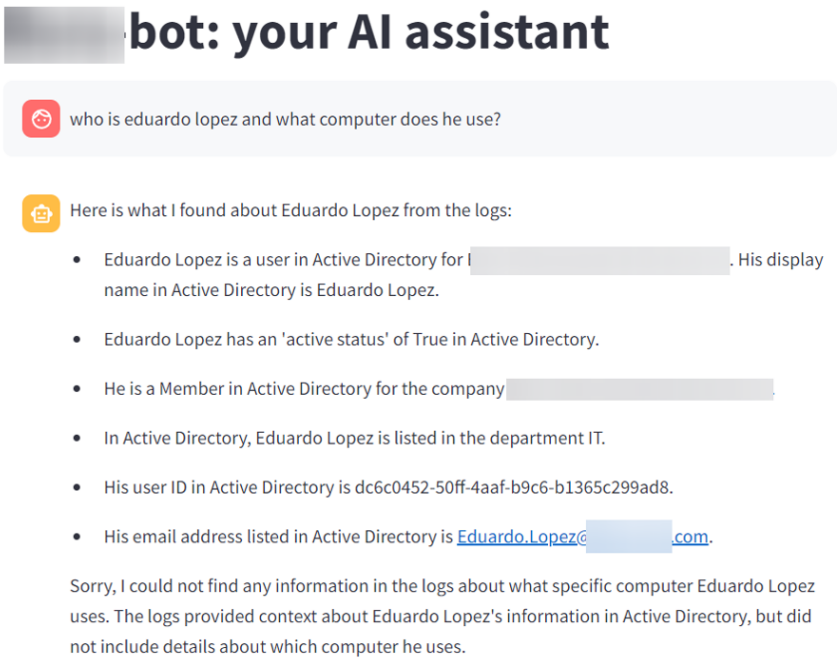


Figure 5.13: Natural language interaction with the AI assistant. Created by E. Lopez.

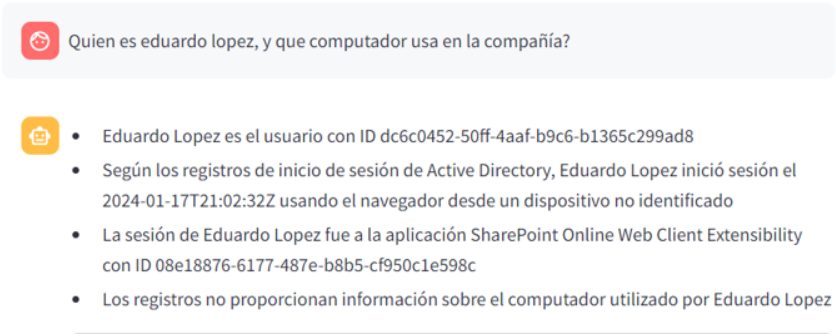


Figure 5.14: Natural language interaction in an alternative language. Created by E. Lopez.

From the information collected from users of the system, the ability to have a conversation with your data generates significant trust and makes adoption remarkably fast. Whereas other systems require training or deep expertise for their use, a chat with an assistant is immediately possible and even available in multiple languages thanks to their pre-training. A similar question to the preceding one was posed in Spanish to the AI assistant. Figure 5.14 shows an example.

The output from the system is grammatically and syntactically correct in the interaction language. The ease of understanding of the system is truly significant. However, it is also important to note that the information provided was different from the preceding example. This is explored next.

Relevance

As with any information system, the information produced by AI needs to be relevant. There are two perspectives that are essential due to the context where this research lies. The first one is the timeliness of the response. In order to be relevant, the information produced needs to be timely (i.e., 'fresh') so it is actionable. If a security analyst retrieves information that is correct, but outdated, the damage from

a malicious attack has probably been inflicted already. The second aspect that is important is that the AI assistant will provide relevant data only if the questions are the correct ones, but even there the system may provide factually-correct yet irrelevant information. The Spanish interaction displayed is a good example of this case. The response from the system was indeed accurate, but what characteristics the system gives to the question "who is ..." came from two different datasets - one from the sign-ins, one from the users.

Security

The AI assistant built for this research required a significant amount of data - a typical requirement in AI. However, an unintended byproduct was highlighted by users: "too much visibility". It is important to note that user access to the original data sources was already in place before the AI implementation took place. However, the extreme ease by which users could retrieve the information took away an existing deterrent: expertise in the data retrieval. Before the AI system, getting the information was difficult, and required specialized skills. The AI assistant dramatically lowered this barrier, making information accessible through a chat conversation. This made the access to using the AI system a critical element design element.

5.4.2 AI system quality

This second construct in TAM pertains to the actual processing mechanics of the system, and how they contribute to the ultimate creation of value [17]. As depicted in Figure 5.11, the information system inherent quality materially influences the intention to use and the user's satisfaction. The following are some of the attributes

pertinent to an AI system quality.

Adaptability

Generative AI significantly disrupts current perceptions about AI. As was described in section 4.4, AI has achieved remarkable results with narrow goals. Although there had been progress towards the use of AI models for tasks other than they were trained for [9], the vast majority of models are optimized for specific uses. However, early indications with Generative AI point to a different future.

This can be best illustrated by the AI assistant created. In its original form, the AI assistant was a conversational chat-bot providing responses based on the parametric knowledge acquired during pre-training. However, using techniques from Retrieval Augmented Generation (RAG) and prompt-engineering, the AI assistant gained capability and accuracy, becoming a good supporting technology. Going beyond semantic search, it has been demonstrated that chain-of-thought prompting improves performance in arithmetic and symbolic reasoning [104]. Based on the early indications, including the ones from this research in cybersecurity, the foundation models such as Large Language Models can be adapted to multiple tasks in very significant manner, much more than traditional information systems.

Availability

As with every other information system, availability is a key attribute required by the organization. The AI assistant deployed runs on hybrid local and cloud infrastructure. It was designed and deployed with availability in mind, while still maintaining the costs contained.

Reliability

Reliability in information systems refers to the consistency of the system in performing its required functions under stated conditions for a specified period of time. This requires that the system is able rapidly scale up/down while maintaining appropriate service levels. Based on the experience with the AI assistant implementation, it became obvious that a mid-size organization would not be able to provision or maintain cost effective infrastructure at an affordable price point. Cloud providers for LLM typically offer services using tokens as the cost unit. A 1,000 word document is approximately 750 tokens - making the use of cloud suppliers cost effective while delivering a very reliable system.

5.4.3 AI service quality

Service quality is a dimension that was added to the model 10 years afters the authors posited the original one. The original TAM had originally three components: the creation of the system (represented through its quality metrics), the use of the system and the consequences of the system's use. With the emergence of information systems as a cornerstone of organizational processes, the focus gravitated away from the information system as a product and towards the complementary service that the IT department provides - in designing, implementing and supporting the system. This dimension of the TAM was posited as an extension to the original model, and released by the authors [18].

Delivering an AI system for a mid-size company required significant technical expertise not readily available in mid-size organizations. Although Generative AI is right now being commercialized heavily across industries, it is still a nascent industry

that requires a resource investment that may be out of reach for many companies. The experience when designing and implementing the AI assistant shed the spotlight on the level of technical expertise needed: although many commercial offerings were used to provide the LLM-capabilities needed, the solution still required significant custom development in programming/scripting languages such as Python and PowerShell. Most mid-size organizations do not have full-stack developers that can create applications using Application Programming Interfaces (API), so it appears that the benefits in the use of AI are still limited to those companies that can invest resources in delivering it.

Once the AI assistant was implemented, it requires minimal support since the User Interface (UI) is so intuitive and easy to use. Most of the direct support required revolves around the data used by the AI RAG, and improving the understanding of the capabilities that the system could offer, and those that it could not.

5.4.4 Intention to use/use of AI

The three quality dimensions influence the following two constructs in the model, impacting the usage of the system. This usage is represented through two dimensions: use of the system and user satisfaction. In a further adjustment of the original model, the **intention to use** was separated from the actual **use**. The first one is an attitude from the user, with the second one being the behavior performed by the user. The relationship between these two elements is causal: an intention to use *causes* the use to take place.

Generative AI is very new and very visible. From a usage perspective, it has been widely reported that ChatGPT had 1 million users within 5 days of the launch.

It is believed that – as of the time this document was created in March 2024 – ChatGPT has 180 million users. It is by most metrics one of the most successful information systems launch ever. However, it is difficulty to separate the novelty from the consistent generation of value for organizations. Most companies are in the process of upskilling their work forces, and looking for use cases that are relevant, practical and with a positive return of investment.

The mid-size company where the AI assistant project was successfully deployed demonstrated that there is a strong appetite for AI. The biggest difficulty is, perhaps, finding the right domain where to apply AI, and also find how to do it within the resource limits of an organization.

5.4.5 User satisfaction and net benefits

The final element in the revised model is net benefits. In the original process model, the information system use led to individual impact, which eventually led to organizational impact – both these categories are collated into the net benefits provided by the system.

The early indications on user satisfaction and net benefits point to a very successful implementation. However, it is too early to understand the mid-to-long term impact of the AI system in the organization. Most of the feedback received has centered on how a technology considered out-of-reach is now available at an acceptable cost for mid-size organizations.

5.5 AI artifacts: Design Science Research

Using the TAM perspective, the specific goals for the AI assistant were explored - the qualities that the 'product' needs to have in order to achieve success in the organization. This section now explores the design process, with the intention of adding valuable information to the knowledge base, thus benefiting practitioners.

Design Science Research (DSR) is fundamentally a problem solving process [1]. Thus, it starts with a problem and produces multiple artifacts in the process of being solved. In the seminal work by Hevner et al. [1], the authors offer seven guidelines to evaluate DSR. I proceed to analyze the posited research through the perspectives of each of the guidelines.

5.5.1 Guideline 1: Design as an artifact

As was explained in section 3, Design Science Research (DSR) in information systems identifies two processes (design and build) and several artifacts (constructs, models, methods and instantiations). The fundamental evaluation of the artifacts is against their utility - whether they solve the problem that originated the design exercise. The process of defining how the problem was to be solved required the creation of multiple artifacts, described next.

Model: the XAI organizational pipeline

As has been described throughout this document, articulating, planning, designing, implementing and supporting AI is very different from those of traditional information systems development. The conceptualization of the process to deliver the solution is

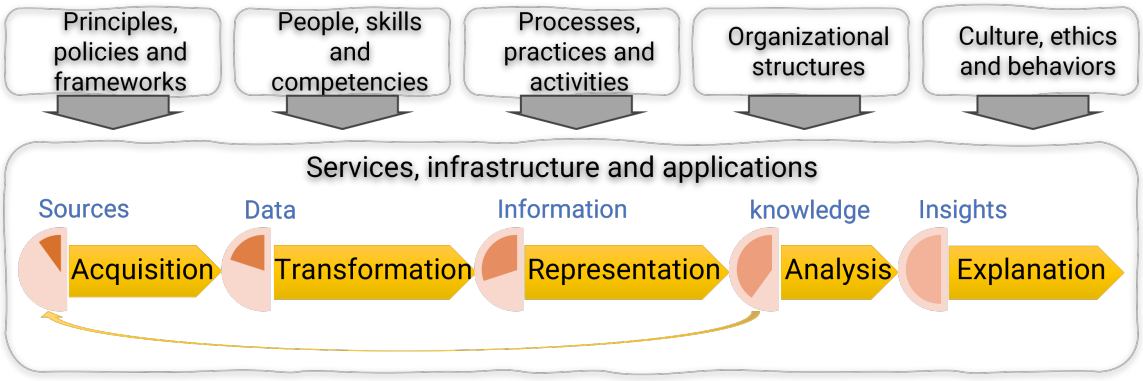


Figure 5.15: Model for AI design: the XAI architecture pipeline. Created by E. Lopez.

a model: the representation of the process. I call this model the XAI architecture pipeline,

The model provides practitioners with a structured guideline to articulate an AI solution. It includes the additional non-technology elements that are essential in driving success. Although beyond the scope of this work, all those elements need their own instantiations in order for the technology solution to be adopted and drive net results.

Under the *services, infrastructure and applications*, the process model for implementation of AI is depicted. Current AI solutions depart from data, or more specifically, from the places where the sources of the data are found. The sources are as diverse as the underlying problem requires. It becomes the first step in the flow that the solution intends to deliver. Sources generate data that may be useful for the ultimate objective that the AI systems intends to fulfill.

Once the sources have been identified, the process of **acquisition** takes place. Different sources may drive different acquisition mechanisms, ultimately oriented towards collecting the data (logically or physically) so the overall solution can use it

in downstream processes. At this stage of the value stream, the sources have been converted into *data*. The data at this stage are raw facts without much context. The data are interpretable within the boundaries of their source, but likely not yet contextualized where value is added.

The next step in the process pertains to the **transformation** of the data into an homogeneous information set. Incremental value is given to the data so it can be compared and contrasted against other data points. Information is data that has been structured with meaning and context, enabling a higher level of capability for the model.

The next step in the XAI architecture pipeline is **representation**. Although the information acquired until this stage has intrinsic value, it is not conducive for use by AI systems. The information needs to be represented in formats that can be processed by technology and thus, further analyze if it is needed.

The sources-to-knowledge conversion happens multiple times, in as many cycles as the problem requires it. In the AI domain, and more particularly in machine learning, knowledge is stored as parameters in a model. The level of complexity of the knowledge stored can be very significant, as it happens with the models such as ChatGPT, with trillion of parameters containing the knowledge the system possesses. This knowledge can be adjusted or augmented through the feedback loop to the sources. If there are new valuable or relevant sources, the process of acquiring them into data, transforming them into information and representing them in the knowledge base allows for the succeeding activities to take place.

Using the knowledge stored through the preceding steps, the **analysis** using

machine learning artifacts can lead to the insights required. The overall pipeline at this point offers the relevant and meaningful elements processed from the knowledge base, and prepares them for delivery to the user or customer.
The final process in the pipeline pertains to the **explanation** of the insights. This is an essential task that seeks explainability and interpretability in the results. The insights may be accurate, correct and relevant, but it is critical that they are delivered in a manner that is transparent and actionable to the user.

The XAI architecture pipeline is a generalization process model for producing a suitable architecture in AI implementations. It enables practitioners to abstract at a higher level the elements that need development, and provides structure to the creation of value through the use of AI. The XAI architecture pipeline shows the different constructs being built through the pipeline, departing from the sources and ending in insights explained to the user. Now I proceed to analyze the model posited in light of another artifact created through this research: the instantiation.

Instantiation: AI assistant for cybersecurity

The conceptual model is applied in the construction of the solution in the case study. The architecture depicted in figure 5.7 reflects the elements explained in the XAI organizational pipeline model. There are multiple sources of data, each with their own dataset going through the **acquisition** stage. Sources included 4 distinct information systems, each exposing their data in different manners and thus, requiring different technology strategies to collect the data. Once the data is centrally stored and accessible, it goes through the **transformation** stage where

it is converted to natural language. The usage of natural language adds meaning and context to the data, producing information. The next stage of **representation** ensures the information is converted to a data type understood by machine learning. In the case study this is done through embeddings, where the information collected from the multiple systems is represented as vectors - allowing for the storing of the knowledge base in an AI-interpretable data type. The output of the stage is now the knowledge base, usable for the analysis that takes place next.

A noteworthy element of the instantiation is the feedback loop that is depicted in the XAI organizational pipeline model. The knowledge base cannot be static. It is the result of the capturing of all users behaviours as represented in each of the participant information systems. It requires permanent updates, to ensure the knowledge base faithfully reflects the regularities of the user behaviours, as well as the novelties and ultimately the anomalies. This feedback loop is implemented in the AI assistant as periodic jobs that take place at least once daily, and that ensure that the knowledge base is current. The AI assistant artifact (i.e., instantiation) performs its **analysis** using as input the knowledge base and the query from a user. Using information from the user's input, the knowledge base is queried for relevant context, which is then provided to a Large Language Model (LLM) that articulates and deliver the **explanation** to the user.

Instantiation: organizational structures; people skills and competencies

There are multiple non-technology instantiations produced in the case study. As was described previously, a centrally coordinated virtual team is formed for the execution

of the processes. The processes associated with the instantiation of organizational structures is beyond the scope of this work, but are critical in ensuring that accountability and responsibility has clear owners. Furthermore, the individuals in charge of performing the work are trained and up-skilled where needed, to maximize the appropriate use of the information system implemented.

Instantiation: principles, policies and frameworks; processes, practices and activities

Multiple processes were designed and implemented to optimize risk and deliver the value sought by the organization. The processes were designed and implemented, with standard operating documentation produced to explicitly codify the practices and activities needed. It is important to note that the audience for the included not only the teams in charge of running the system, but also the employee population as a whole. In particular, the policy of *acceptable use of IT resources* ensured wide communication was delivered, managing expectations of the organization around electronic monitoring and complying with the regulatory mandates issued by the province of Ontario.

Instantiation: culture, ethics and behaviours

The AI solution described in this research is a fundamental part of a holistic strategy oriented towards improving the cybersecurity posture of the organization. Using training and purposeful communication, a culture of constant vigilance and careful use of IT resources was encouraged, a fundamental part of the cybersecurity strategy. An example of a successful instantiation was the delivery of cybersecurity trick

phishing emails that gauged the preparedness of the employee population. Using the results (e.g., percentage of clicks) from each drill, broad communication to the organization was performed, with the ultimate objective of influencing culture and behaviours.

This section explored how multiple design artifacts were created as part of this research. A model for XAI architecture with data-centric constructs crystallizing the multiple stages and intermediate outputs. In addition to this, an instantiation of the AI artifact in alignment with the model was created and described in detail. The final set of artifacts were non-technological in nature, and included the organizational structures and standard operating documentation that are essential in achieving the objective of enhanced organizational cybersecurity posture.

5.5.2 Guideline 2: Problem relevance

Design Science Research has as fundamental objective the development of technology based solutions for relevant and important problems [1]. This research approaches two distinct, yet critical problems that are current, relevant and critical for organizational success. The dawn of Generative AI (GenAI) is generating remarkable excitement in industry and academia – albeit without the supporting processes readily available for traditional systems (e.g., waterfall software development life cycle). Designing, planning, implementing and supporting AI systems requires a new set of guidelines and best practices that are still being developed. This is an important problem to address and the ultimate objective of the research.

Furthermore, the case study for the use of AI in this research tackles a very significant challenge for every organization using information systems. Improving the cybersecurity posture of a mid-size organization is an important objective that is richly developed throughout this work.

5.5.3 Guideline 3: Design evaluation

Multiple different methods can be used in the evaluation of design. This research used extensively two types. The first was observational - using the model posited, a complete case study was undertaken, departing from a business need and ultimately implementing a system in productive use. Adoption of the system has been successful, notwithstanding the many improvements suggested by users of the system. This is a normal expectation on new systems, especially as they implement radically new approaches.

A second, and perhaps more rigorous evaluation of the design artifacts was performed by addressing a different business need. A proof of concept for the creation of an AI assistant was delivered in parallel to the cybersecurity implementation, and is described next.

Business problem

Most regulated organizations – such as the one in the case study – are required to maintain a set of documentation that is considered 'controlled'. Controlled documents are usually standard operating documents that describe formally myriad processes performed in the organization. A controlled document goes through a very strict life cycle that includes drafting, approval, and eventually, training. A mid-size

organization such as the one in the case study may have thousands of documents – approximately 9,000 in this specific case. This corpus basically captures how the organization operates, but is very extensive and difficult to query.

Use of the XAI organizational pipeline

Using the model posited as one of the design artifacts in this research, a system was designed and implemented to act as an AI assistant for controlled documentation. The architecture followed the model as follows:

Acquisition The controlled document repository was used as the fundamental source. Other sources exist, but to keep the proof of concept contained, only this source was employed. The resulting set was centrally stored and now becomes the *data* in the XAI organizational pipeline.

Transformation Controlled documents were stored in multiple formats. Most of them were PDF files, but also included documents created by the Microsoft Office package. Once the documents were transformed, a single corpus of text-based files was created, following the requirements for downstream processing. The original 9,000 documents were converted to approximately 350,000 text files - what is termed *information* in the XAI organizational pipeline.

Representation Each of the files produced was embedded and stored as vectors in a purpose-built database. By doing this, the *knowledge* on how to the company operates was codified and available for querying.

Analysis The AI assistant for controlled documentation was implemented so it answered questions by users through a chat interface. Using Retrieval Augmented Generation (RAG) as the technical architecture, the questions from the user were employed in a semantic search against the knowledge base. Context and user input were put together and delivered to the final stage of the process.

Explanation The final stage involves a Large Language Model (LLM) that uses query and context to produce an interpretable output to the user.

The successful application of the XAI organizational pipeline process to a different use case reinforced the utility and value of the design artifact created. Evaluation under a different set of circumstances and dynamics strengthens the value proposition posited.

5.5.4 Guideline 4: Research contributions

The most important contributions of this work pertains to the design artifact. The model, constructs and instantiations – successfully used in both cybersecurity and controlled documentation – can create significant value to both academia and industry. AI, and in particular Generative AI, is a remarkable technology advance that needs and creates very unique dynamics that need to be explored, and are thoroughly described in this research.

5.5.5 Guideline 5: Research rigor

Rigor has a direct dependency with the effective use of the knowledge base [1]. The phenomena explore in this research required the use of myriad concepts from different domains. The core knowledge base pertained to AI. Over the course of an extended period of time, state-of-the-art technologies were used to fundamentally deliver a threat detection artifact. Although GenAI is the current instantiation, preceding artifacts included Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Transformers. Each of these developments required in-depth use of the mathematical formulations in alignment with best practices in industry and academia.

The last iteration of the research is captured in this document - using the latest technology advances in Large Language Models (LLM), but rounding it with essential elements for structured delivery of information systems. A de-facto standard in IT governance acted as the backdrop against which a complete solution was rigorously formulated. The XAI organizational pipeline model allows for the creation of a holistic structure that centers around an AI artifact, but must also include organizational structures with skilled individuals following a predefined set of processes, practices and activities. Principles, policies and frameworks are instantiated and communicated, strengthened by culture and behaviours enacted.

5.5.6 Guideline 6: Design as a search process

Detecting potential threats taking place in an organization is a difficult task. There is no definite solution that can unequivocally identify that there is a malicious actor inflicting damage through an information system. The process by which an optimal solution was found, designed and instantiated required a logical process of elimination

where constraints and resources needed to be explicitly formulated, and adequate solutions implemented, tested and chosen or discarded.

Using a case study enabled the research to evaluate the design in a real-world setting, fueling myriad iterations to find optimal solutions. Although the artifacts documented in this work may appear close to final, reality is that they were part of a search process that can continue indefinitely. Furthermore, the design posited is less a mandate on how to implement AI in an organization, and more a departing point that practitioners can use to better educate their design decisions and potential strategies to achieve value in an organization.

5.5.7 Guideline 7: Communication of research

The nature of knowledge in this design science research was eclectic. Although technology played a preponderant role, it was far from the only knowledge base used. The holistic approach taken in the research is fundamental to success in an information system deployment. Because of this approach, the audience for this research includes not only technology practitioners, but also managers in organizations where the technology can create value. Although it may be important to understand the technological underpinnings, it is, perhaps, as critical to foster a realization on other key factors that affect organizations. For managers it may be more important to understand the conditions and constrains under which a solution such as the one posited in this research can be adopted.

To improve the understanding of the instantiation produced, the appendices to this document include key excerpts of the programming employed, a best practice recommended for design science research [1].

Chapter 6

Conclusion

Implementing an Artificial Intelligence (AI) system is fundamentally different from traditional information system implementations. Although some success factors are common across technology initiatives, practitioners should consider AI's particular dynamics to maximize utility, optimize risk and realize its benefits.

6.1 Common problem and solution space in AI

As it is the case with information system projects in organizations, having clear objectives and a well-articulated use case is essential to success. The current AI technologies are able to address problems where the fundamental need is prediction or classification, and where there is sufficient data to warrant its use. In the case of Generative AI, the problem space is narrower - as the strength of the technology relies on creation of content. The case study undertaken for this research provided key insights on potential applications of the technology. The XAI organizational pipeline can be used as a reflection of the solution space. Raw data progresses through a

pipeline until it reaches a state where it can be considered a *knowledge base*. From the knowledge acquired it is possible to perform further analysis that create the value sought by a practitioner.

As was demonstrated through the case study, raw data from multiple sources were acquired, transformed and ultimately represented in a user behaviour model stored as embeddings. From that point forward, the analysis focused on identifying potential anomalies by comparing recent user behaviour with the baseline knowledge. The other use case explored as an application of the XAI organizational pipeline followed the structure well. Using as source the controlled documents repository, raw data was acquired, transformed, represented and stored as vector embeddings. As a knowledge base, it was then possible to use it in analysis, such as asking the AI assistant for information based on the knowledge stored. Although it may be obvious, it is important to emphasize the importance of choosing the right problem to solve with AI. The very public explosion of interest in AI has created a perception – evident in the case study's company – that it can solve many problems that are still outside the realm of possibility. A good fit between the problem and solution space is essential to the success of an AI initiative.

6.2 IT skills in the era of AI

The case study highlighted the mismatch currently in place between the skills required (especially in IT) and the needs driven by an AI initiative. The focus of the case study – a mid-size organization with approximately 15 people in IT – was not well suited for an AI initiative. The following are some of the observations gleaned throughout the case study.

AI and Data science Although the level of abstraction in the AI commercial offerings has increased, there is still a fundamental need for the conceptual foundations underpinning data. The mid-size, health sciences organization where the case study took place had limited understanding of core data science concepts, presenting challenges when explaining how AI was able to provide its results. Furthermore, the delivery of an AI artifact requires proficiency in concepts that are very foreign to traditional systems development. Training, fine-tuning, embeddings are just some of the components that need thorough understanding so they are applied suitably to an organizational use case.

Data and software engineering The case study confirmed that AI initiatives must include core data tasks such as Extraction, Transformation and Load (ETL), long a practice in traditional IT departments. However, the ability to orchestrate the multiple intertwined components that are part of an AI solution drove a significant need for software engineering expertise, which is not readily available in most mid-size organizations. In order to deliver value, AI needed to be integrated in the fabric of the organization's information landscape. As of the time of this research, AI deployments are not yet equivalent to a Commercial-Off-The-Shelf (COTS) software that organizations can procure, and thus require an implementation closer to custom software development.

Business domain knowledge The essential raw material for AI – data – requires significant understanding of the fundamental processes associated with the data flows in the organization. The case study was rooted in the need to improve the organization's cybersecurity posture. Without strong domain knowledge it would have been

not possible to design an AI-based solution that was fit-for-use or fit-for-purpose. The additional use case explored - AI for controlled documentation – also needed strong knowledge about the unstructured data and the repositories in which they are stored. Notwithstanding the remarkable advance of AI, the case study demonstrated that domain knowledge is still a critical element in project success.

6.3 AI Project management

Scope, timeline and cost are considered the three constraints most project managers need to control [53]. Multiple relationships can be interpreted based on the triple constraint. For example, an increase in project scope may lead to an increase in the time needed for completion. A project schedule can be 'crashed' (i.e., shortened) if additional cost is incurred. Although considered by some authors to be overly simplistic [98], this model is the cornerstone for project management practitioners. However, AI brings new elements that are important to consider.

Unlike many other information systems, AI outcomes are far less predictable. Whereas a business intelligence project clearly defines the metrics to be calculated or presented, AI outputs are inherently uncertain. This case is specially poignant in the case of Generative AI, where hallucinations are currently considered unavoidable. Furthermore, the iterative process by which AI projects are delivered may create significant uncertainty in scope and timeline, creating serious challenges for practitioners. Based on the experience in the case study, an AI implementation can optimize risk by delivering in three stages. First, a Proof of Concept (PoC) that demonstrates the potential value of AI when applied to a narrow scope. If the PoC is deemed successful, proceeding to a pilot deployment allows project managers and IT practitioners

to extend the perceived value achieved in the PoC to a IT production environment. If the pilot is successful, a full-scale AI implementation may be pursued.

Although this three-stage process does not eliminate the uncertainty with respect to AI deployments, it manages risks by reducing failure cost. A PoC may conclude that the solution chosen is not fit-for-purpose for the problem, or fit-for-use for the organization. Exiting the project at this stage is a potential outcome that can still be considered a success as it avoids wasting of resources.

6.4 AI affordability

Research and implementation in AI are very expensive propositions. The requirements in purpose-built hardware (such as Graphical Processing Units GPUs) are significant and require investments beyond the reach of most organizations. In terms of human resources the situation is similarly challenging. In 2011, the proportion of AI PhDs working in industry (40.9%) was roughly equal to the one working in academia (41.6%). However, as recently as 2021 the number headed for industry increased to 65.4%, a trend that appears to be accelerating [94]. Given the dynamics in both infrastructure and talent, AI initiatives can be a very expensive proposition, limiting the opportunity for small-to-mid size companies to compete.

However, the particular characteristics around Generative AI may present an opportunity for organizations of all sizes and types. The majority of the cost in a GenAI initiative would be the pre-training of the model (i.e., training 'from scratch'). Once a model is created, it can be used and adjusted for a fraction of the cost. The case study allowed visibility on this dynamic.

The organization where the AI assistant was implemented is a mid-size organization.

It does not have the resources to maintain a high-performance environment to train a Large Language Model (LLM). Although the information is proprietary, several news outlets speculated that the cost of training GPT-4 from scratch was around US$ 100 million - out of reach for most corporations. Thus, the only affordable option was to leverage a pre-trained model and adapt it for the use case required. This is now possible because large Infrastructure-as-a-Service (IaaS) providers such as Amazon and Google have implemented AI technologies that can be cloud-provisioned. The trends in AI infrastructure closely resemble the creation of the IaaS market by Amazon - the capability of commercializing on-demand infrastructure.
Multiple services are used for the case study, interfaced via custom code. The following is the cost estimates for the AI project, and its ongoing maintenance.

Initial configuration The processes of acquisition, transformation and representation in the XAI organizational pipeline required infrastructure and LLM use for the calculation and storage of the knowledge base (i.e., embeddings). The infrastructure included compute and storage with a cost of approximately US$4,000. For the embedding process, the most cost-effective model is used (i.e., AWS Titan), with 1,000 tokens priced at US$0.0001 at the time this research. The initial load of 10 million documents (i.e., vectors) embedded had a cost of approximately US$ 1,000. In total the initial configuration and setup of the system took place over 2 months, for a total cost of approximately US$10,000.

On-going operation The system operation involved the utilization of compute and storage, and additionally the LLM for the interaction with the user. The infrastructure costs are approximately US$4,000. For the processes associated with the user

interface, the model selected was Anthropic's *Claude instant*, which costs US$0.00248 per 1,000 input tokens and US$0.00838 for 1,000 output tokens. Based on the current prompt structure, including the context provided as part of the Retrieval Augmented Generation (RAG) employed, a single question/answer by the security analyst would cost approximately 1.5 cents.

The total cost of the implementation was palatable as a proof of concept for a mid-size organization. Proceeding to the pilot stage is probable given the learning curve already traversed on the new technology, and the relative modest cost of operation.

6.5 AI composability

In information system design, composability can be defined as architecture principle where multiple components can be selected, connected or combined, ultimately creating complex structures in support of the system's objective. Some vendors refer to this design as a micro-services architecture [105], with individual components performing functions in relative independence from other components.

The maturing of AI as a technology has somewhat mimicked that of software development. In the early stages the architecture was monolithic - few or single block of code enabling the capabilities of the software. However, Generative AI has gravitated towards increased commoditization, where LLMs can be used in multiple parts of the software flow.

The final architecture for the system evolved as constraints and/or opportunities were found. A micro-services software architecture enabled the interfacing with very diverse data sources, yet using a consistent approach. The compute required to run the system was limited - thanks to the use of 3rd party services accessed

via APIs. The network needs for the solution were significant, but adequately managed within the regular constraints of capacity and capability. No incremental hardware was needed to enable the network connectivity. The most important learning pertained to storage. As the project progressed, it became evident that the solution necessarily would consume significant resources to store the data – a direct result of the RAG-centric implementation. The initial load of the data in the system created approximately 250GB over $\tilde{1}0$ million documents - necessarily driving two dynamics: system response and cost. Fortunately, the costs of storage has decreased significantly over time. The current Amazon service (Simple Storage Service or S3) is currently priced at US$ 0.023 per GB per month, up to 50TB. That translates to approximately US$ 6 in the initial month, growing at approximately 30% per month until a cutoff date is configured in the system.

6.6 Potential research streams

AI is potentially the defining technological advance of this generation. The case study enabled the rich exploration of multiple facets that a mid-size organization faces when deciding to embark on AI initiatives. The complexity of an information system project can dramatically increase when a novel technology is used, bringing new risks that practitioners must address. The work delivered through this research, and captured in this document, point to an exciting future that researchers must continue to develop in order to optimize risk, resources and realize value.

From an information systems perspective, a potential research stream can revolve around the XAI organizational pipeline model - one of the design artifacts produced

in this document. The present work posits a potential tool for practitioners to conceptualize, architect and deliver AI-powered projects. Research to verify the validity and usability of the model across multiple diverse AI projects, under different organizational constraints and environments would be a significant contribution to the information systems field.

The case study analyzed provided a perspective on the different elements driving adoption and resulting in net benefits for the organization. The testing and verification of the Technology Acceptance Model (TAM) under AI can also provide a fascinating research stream that would assist practitioners in the delivery of value with the use of an era-defining technology.

Appendix A

Key functions in Python

In alignment with the practices in Design Science Research (DSR), key technical elements are included in the appendix of the document for the benefit of practitioners [35]. Given the lenght of the code base (approximately 3,000 lines of code in Python and PowerShell), only key elements of the code base are documented in the Appendix.

Instantiation of clients There are two key services that are used from the provider AWS. *OpenSearch Serverless* is the service used for the storage and retrieval of embeddings. It is a vector database based on the tool FAISS. The second service is *Bedrock*, which is used for the purposes of LLM interaction. This includes the creation of embeddings as well as the User Interface (UI).

Listing A.1: Instantiation of clients

```
1    def create_clients_v2 (key,secret ,aoss_endpoint):
2        from requests_aws4auth import AWS4Auth
3        from opensearchpy import OpenSearch, RequestsHttpConnection
4        import boto3
```

```
5       import os
6       from dotenv import load_dotenv
7
8       load_dotenv()
9       """
10      This function is used to create both the Bedrock client and the OpenSearch client
11      :param key: Unique client/application key created in AWS
12      :param secret: Client secret
13      :param aoss_endpoint: is the OpenSearch Serverless endpoint, i.e., a single collection
14      :return: bedrock client, OpenSearch client
15      """
16      service = 'aoss'
17      awsauth = AWS4Auth(key, secret, os.getenv('region'), service)
18      # IMPORTANT: Remember to check that the Data policy for the collection in AOSS has
    as one of the principals \
19      # the Bedrock-API-Admin-Access user created in IAM.
20      openSearchClient = OpenSearch(
21      hosts = [{'host': aoss_endpoint, 'port': 443}],
22      http_auth = awsauth,
23      use_ssl = True,
24      verify_certs = True,
25      http_compress = True,
26      connection_class = RequestsHttpConnection,
27      timeout=60
28      )
29      bedrockClient = boto3.client('bedrock-runtime', os.getenv('region'), endpoint_url=os.
    getenv('bedrock_endpoint'))
30      return bedrockClient, openSearchClient
```

API call for embedding The code below is a general function called whenever there is a need to embed text into a dense vector.

Listing A.2: Embedding of text using Bedrock

```
1   def get_embedding_v2(userInput, bedrockClient):
2       """
3       This function is used to generate the embeddings for each question the user submits.
4       :param body: This is the question that is passed in to generate an embedding
5       :return: A vector containing the embeddings of the passed in content
6       """
7       import json
8       # defining the embeddings model
9       modelId = 'amazon.titan-embed-text-v1'
10      accept = 'application/json'
11      contentType = 'application/json'
12
13      # creating a json from the user's input
14      body = json.dumps({"inputText": userInput})
15
16      # invoking the embedding model
17      response = bedrockClient.invoke_model(body=body, modelId=modelId, accept=accept,
    contentType=contentType)
18
19      # reading in the specific embedding
20      response_body = json.loads(response.get('body').read())
21      embedding = response_body.get('embedding')
22      return embedding
```

From sources to data: acquisition The XAI organizational pipeline departs from multiple sources of data that are acquired. The following code creates the documents for one of the sources: Microsoft Graph.

Listing A.3: Acquisition of Microsoft Graph sources

```
1  def create_msgraph_docs_v2(folder):
2      initialize_msgraph_docs(folder)
3      get_msgraph_users(folder)
4      get_msgraph_signins_v2(folder,2)
5      get_msgraph_devices(folder)
6      get_msgraph_directoryaudits_v2(folder,2)
7      get_msgraph_registrations_v2(folder,2)
```

As can be observed, the MS Graph source includes 5 distinct datasets that are acquired. Each of the lines calls a specific Python function to perform the acquisition. As illustration, the Microsoft Graph sign-ins are acquired as follows.

Listing A.4: Acquisition of Microsoft Graph Sign-ins

```
1  def get_msgraph_signins_v2(folder, days):
2      import pandas as pd
3      import os
4      import datetime
5      import requests
6      import json
7      import msal
8      import dotenv
9      from dotenv import load_dotenv
10     from datetime import timedelta
11
```

```
12          load_dotenv()
13          token=(get_token(os.getenv('MS_Graph_tenant_id'),os.getenv('MS_Graph_client_id'),os.
        getenv('MS_Graph_client_secret')))
14
15          # Calculate the date range for the preceding 5 days
16          end_date = datetime.datetime.now().date()
17          start_date = end_date - timedelta(days=days)
18          dateString = start_date.strftime('%Y-%m-%dT%H:%M:%SZ')
19
20          # Now directory registrations
21          json_data=make_api_request_v2('https://graph.microsoft.com/v1.0/auditLogs/signIns?
        $filter=createdDateTime ge ' + dateString,token)
22          df=pd.DataFrame.from_dict(json_data['value'])
23          df = df.fillna('')
24          df['sentence']="Active Directory recorded the user signing id " + df['id'] + " for the
        user " + df['userDisplayName'] + " (with 'user ID' " + df['userId'] +\
25          ") signed on " + df['createdDateTime'] + " to the application " + df['appDisplayName']
        + " (with 'app id' " + df['appId'] + "). The user used as client " +\
26              df['clientAppUsed'] + " from IP address " + df['ipAddress'] + " using the resource "
        + df['resourceDisplayName'] + " (with 'resource id' " +\
27                  df['resourceId'] + ") at location " + df['location'].astype(str) + " and using
        the device described as " + df['deviceDetail'].astype(str) + \
28                      ". The status for the sign-in event is described as " + df['status'].astype(
        str) + "."
29          df['createdDateTime']=pd.to_datetime(df['createdDateTime'],format='mixed')
30          df['Date']=df['createdDateTime'].dt.date
31          for TheDate in df['Date'].unique():
32              dfToTxt = df[df['Date']==TheDate]['sentence']
```

```
33              dfToTxt.to_csv(folder + '/user_signin_docs_' + str(TheDate) + '.txt',index=False,
    header=False)
```

From data to information: transformation This code exemplifies how the data is converted to information (i.e., documents). Each original file is 'chunked' into documents of 1,000 characters, which are then embedded and uploaded to the index in the vector database.

Listing A.5: segmenting documents and upload to index in vector database

```
1   def cybersecurity_load_msgraph_folder_to_index_v2(folder, bedrockClient, OpenSearchClient,
    index):
2       from langchain.text_splitter import RecursiveCharacterTextSplitter
3       from langchain.document_loaders import TextLoader
4       import json, os, shutil
5       from dotenv import load_dotenv
6       load_dotenv()
7
8       print("Loading MSGraph files from " + folder + " to index in collection." + str(
    OpenSearchClient))
9        text_splitter = RecursiveCharacterTextSplitter(
10          chunk_size=1000, # dense documents, small chunks
11          chunk_overlap=100,
12      )
13      # first, count the number of files to process
14      countFiles =0
15      for path in os.scandir(folder):
16          if path.is_file():
17              countFiles += 1
```

```
18          print("total of " + str(countFiles) + " files to process.")
19          fileNumber=0
20          totalPages=0
21          totalChunks=0
22          for file in sorted(os.listdir(folder)):
23              fileNumber += 1
24              filename = folder + "/" + os.fsdecode(file)
25              # loader=PyMuPDFLoader(filename,extract_images=True) # this loader extracts
        texts from images. very slow and not too accurate.
26              loader=TextLoader(filename) # this loader only extracts text
27              try:
28                  documents = loader.load()
29                  print('Loaded document ' + filename)
30              except:
31                  print('Error while loading, skipping file : ' + filename)
32              pass
33
34              # Performing the splitting of the document(s)
35              doc = text_splitter.split_documents(documents)
36              print('Splitted document ' + filename)
37              # getting metrics for the file
38              totalPages = totalPages + len(documents)
39              totalChunks = totalChunks + len(doc)
40              print("Processing file " + filename + ":" + str(fileNumber) + "/" + str(countFiles
        ) + ". Generated " + str(len(documents)) + " pages with " + \
41                  str(len(doc)) + " chunks. Cumulative pages: " + str(totalPages) + ",
        cumulative chunks: " + str(totalChunks))
42              if filename.split('/')[3].startswith("device_info_docs_"):
43                  dateLog = file.split('device_info_docs_')[1].split('.txt')[0]
```

```
            source = 'MSGraph_Device'
        elif filename.split('/')[3].startswith("directory_audit_event_docs_"):
            dateLog = file.split('directory_audit_event_docs_')[1].split('.txt')[0]
            source = 'MSGraph_DirectoryAudit'
        elif filename.split('/')[3].startswith("user_info_docs_"):
            dateLog = file.split('user_info_docs_')[1].split('.txt')[0]
            source = 'MSGraph_User'
        elif filename.split('/')[3].startswith("user_registration_docs_"):
            dateLog = file.split('user_registration_docs_')[1].split('.txt')[0]
            source = 'MSGraph_Registration'
        elif filename.split('/')[3].startswith("user_signin_docs_"):
            dateLog = file.split('user_signin_docs_')[1].split('.txt')[0]
            source = 'MSGraph_SignIn'
        docNumber = 0
        # now we process each chunk
```

From information to knowledge: Representation Once the information is captured in the text documents, the next step in the process pertains to the representation of the information in embeddings - what can be described as parametric knowledge since all the information is stored in the embeddings as per the parameters of the model.

```

        for i in doc:
            try:
                # The text data of each chunk
                chunkContent = i.page_content

                jsonChunk = json.dumps({"inputText": chunkContent})
```

```
8                       chunkEmbedding = get_embedding_v2(jsonChunk,bedrockClient)
9                       text = chunkContent
10                      vectors = chunkEmbedding
11                      indexDocument = {
12                      'borarag-cybersecurity-vectorfield': vectors,
13                      'text': text,
14                      'borarag-cybersecurity-date' : dateLog,
15                      'borarag-cybersecurity-source' : source
16                      }
17                      response = OpenSearchClient.index(index=index,body=indexDocument,
    refresh=False)
18                      print("File " + filename + "(No. " + str(fileNumber) + " of )" + str(
    countFiles) + "), Chunk " + str(docNumber) + " of " + str(totalChunks))
19                      docNumber += 1
20                      # print(f"page: {i}") # not printing to the output as it is too much data
    and kills vscode
21                  except Exception as error:
22                      print(error)
23                      print("Error while indexing in file " + filename + ". Skipping.")
24                      pass # move to the next document
25              print('Finished indexing the file ' + filename + ', moving it to the processed
    folder.')
26
27              # Check if the file already exists in the processed folder and delete it if it does
28               file_to_delete = 'Data/Processed/MSGraphDoc/' + filename.split('/')[3]
29              if os.path.exists( file_to_delete ):
30                  os.remove( file_to_delete )
31                  print('Existing file ' + file_to_delete + ' has been deleted.')
32              shutil.move(filename, 'Data/Processed/MSGraphDoc')
```

```
33              print('File has been moved to the processed folder')
```

Multiple checks are performed to ensure the information is not re-indexed. THe following code snippet shows the ingestion process that is called for the browsing history.

```
1   def cybersecurity_ingest_browsinghistory_v3(folder, processed_folder):
2   """
3   This function performs all the processes for ingestion. First it creates the clients, then
4   loops through the files in the folder and:
5   - Checks whether the file exists in the index and if so, deletes all chunks
6   - Uploads and indexes each of the files in the folder
7   - Deletes the file after processing.
8   """
9   import os, filecmp
10  from dotenv import load_dotenv
11  import os
12  load_dotenv()
13
14  bedrockClient, OpenSearchClient = create_clients_v2( os.getenv('AWS_ACCESS_KEY_ID'),os.
    getenv('AWS_SECRET_ACCESS_KEY'),os.getenv('opensearch_host_RAGCybersecurity'))
15
16  if os.listdir(folder):
17      for file in sorted(os.listdir(folder)):
18          file_with_path =os.path.join(folder, file)
19          print(file, file_with_path)
20
21          if cybersecurity_check_same_document_in_processed(file_with_path,os.path.join(
    processed_folder, file)):
```

```
22                # if file is the same as the one in the processed folder, delete it since it
    was already indexed
23                os.remove(file_with_path)
24                print(file + ' has been deleted from the Pending folder.')
25            else:
26                # if file is different or does not exist in the processed folder, let's find out
     if it has chunks
27                print('File ' + file + ' does not exist or is different from the one in the
    processed folder. Proceeding to review if it exists in the index.')
28                if cybersecurity_check_document_in_index_v2(OpenSearchClient,file_with_path, os.
    getenv('vector_index_name_RAGCybersecurity_v2')):
29                    print('File ' + file + ' is in the index, proceeding to delete its chunks
    before reindexing.')
30                    cybersecurity_delete_document_chunks_v2(OpenSearchClient,file_with_path, os.
    getenv('vector_index_name_RAGCybersecurity_v2'))
31                print('Chunks for file ' + file + ' have been deleted. Proceeding to index the
     file .')
32            cybersecurity_load_browsinghistory_folder_to_index_v2(folder, bedrockClient,
    OpenSearchClient,os.getenv('vector_index_name_RAGCybersecurity_v2'))
33
34        print('Finished browsing history ingestion!')
```

From knowledge to insights: Analysis Once the information is stored as parametric knowledge, it is possible to perform analysis. The following code demonstrates how the information is extracted from the parametric knowledge base.

```
1    def get_all_vectors(source,index):
2    """
3    This function creates the clients for OpenSearch and Bedrock,
```

```
4       searches the index for all chunks of a given source and returns them
5       in a dataframe.
6
7       Args:
8           source (string): potential sources: 'BrowsingHistoryView', 'CrowdStrike_Host', '
        CrowdStrike_Login', 'MSGraph_Device',
9                           'MSGraph_DirectoryAudit', 'MSGraph_Registration', 'MSGraph_SignIn', '
        MSGraph_User', 'OpenVAS_Host', 'OpenVAS_HostMax', 'OpenVAS_Results'
10      Returns:
11          df: _description_
12      """
13      import os, dotenv
14      import pandas as pd
15      from dotenv import load_dotenv
16      load_dotenv()
17      bedrockClient, OpenSearchClient = create_clients_v2(os.getenv('AWS_ACCESS_KEY_ID'),os.
        getenv('AWS_SECRET_ACCESS_KEY'),os.getenv('opensearch_host_RAGCybersecurity'))
18
19      query = {"query" : {"bool" : {"must" : [{"match": {'borarag-cybersecurity-source' : source
        }}]}},"sort": [{ "_id": "asc" }]}
20
21      totalPages = 0 # initialize
22      hits = [] # initialize
23      response = OpenSearchClient.search(index=index, size=10000, body=query)
24      pagesLeft = True
25      while pagesLeft == True:
26          totalPages = totalPages + 1
27          hits = hits + response['hits']['hits']
28          print('Found ' + str(len(hits)) + ' total hits after page ' + str(totalPages))
```

```
29
30          # now let's find out if there are more pages to go through
31          search_after = hits[-1]['sort'] # get the last in the list
32          print('search_after: ' + str(search_after))
33          query["search_after"] = search_after # modify the query
34          print(query)
35          response = OpenSearchClient.search(index=index, size=10000, body=query)
36          newHits = response['hits']['hits']
37          if len(newHits) == 0:
38              pagesLeft = False
39
40      print('Found a total of ' + str(len(hits)) + ' chunks for ' + source + '.')
41      df = pd.DataFrame(hits)
42      df['vector'] = df['_source'].apply(lambda x: x['borarag-cybersecurity-vectorfield'])
43      df['date'] = df['_source'].apply(lambda x: x['borarag-cybersecurity-date'])
44
45      return df
```

Using the vectors acquired, a Principal Components Analysis (PCA) is performed to avoid the curse of dimensionality and streamline the processing of data.

```
1   from sklearn.decomposition import PCA
2 import pandas as pd
3 # the maxiuym number of components is 1536, as that is the dimension of the vectors, but we
      may not have that many
4 # so we need to check the length of the vectors and use the minimum of that and 1536
5
6 pca_BrowsingHistoryView_vectors = PCA(n_components=min(len(BrowsingHistoryView_vectors)
      ,1536))
7 print('Created the PCA instance for BrowsingHistoryView')
```

```
8
9  pca_CrowdStrike_Host_vectors = PCA(n_components=min(len(CrowdStrike_Host_vectors),1536))
10 print('Created the PCA instance for CrowdStrike_Host')
11 pca_CrowdStrike_Login_vectors = PCA(n_components=min(len(CrowdStrike_Login_vectors),1536)
       )
12 print('Created the PCA instance for CrowdStrike_Login')
13 pca_OpenVAS_Host_vectors = PCA(n_components=min(len(OpenVAS_Host_vectors),1536))
14 print('Created the PCA instance for OpenVAS_Host')
15 pca_OpenVAS_HostMax_vectors = PCA(n_components=min(len(OpenVAS_HostMax_vectors)
       ,1536))
16 print('Created the PCA instance for OpenVAS_HostMax')
17 pca_OpenVAS_Results_vectors = PCA(n_components=min(len(OpenVAS_Results_vectors),1536))
18 print('Created the PCA instance for OpenVAS_Results')
19 pca_MSGraph_Device_vectors = PCA(n_components=min(len(MSGraph_Device_vectors),1536))
20 print('Created the PCA instance for MSGraph_Device')
21 pca_MSGraph_DirectoryAudit_vectors = PCA(n_components=min(len(
       MSGraph_DirectoryAudit_vectors),1536))
22 print('Created the PCA instance for MSGraph_DirectoryAudit')
23 pca_MSGraph_signin_vectors = PCA(n_components=min(len(MSGraph_signin_vectors),1536))
24 print('Created the PCA instance for MSGraph_SignIn')
25 pca_MSGraph_Registration_vectors = PCA(n_components=min(len(
       MSGraph_Registration_vectors),1536))
26 print('Created the PCA instance for MSGraph_Registration')
27 pca_MSGraph_user_vectors = PCA(n_components=min(len(MSGraph_user_vectors),1536))
28 print('Created the PCA instance for MSGraph_User')
```

At this point it is possible to perform an anomaly detection with isolation forest.

```
1     def get_anomaly_indices(vector, contamination):
2     """
```

```
    This function takes a vector and a contamination rate and returns the indices of the
    anomalies
    :param vector: the vector to be analyzed
    :param contamination: the contamination rate
    :return: the indices of the anomalies and the score factors
    """
    from sklearn.ensemble import IsolationForest
    # Create an instance of the IsolationForest model
    isolation_forest = IsolationForest(contamination=contamination)
    isolation_forest.fit(vector.tolist())
    anomaly_predictions = isolation_forest.predict(vector.tolist())
    anomaly_indices = np.where(anomaly_predictions == -1)[0]
    score_factors = isolation_forest.score_samples(vector.tolist())
    print('Found ' + str(len(anomaly_indices)) + ' anomalies.')
    return anomaly_indices, score_factors
```

The information gathered up to this point is then passed to the LLM using prompt engineering, as shown below.

```
def answer_query_v4_cybersecurity(user_input,bedrockClient,opensearchClient,index,
    vector_field):
    """
    This function takes the user question, creates an embedding of that question,
    and performs a KNN search on your Amazon OpenSearch Index. Using the most similar
    results it feeds that into the Prompt
    and LLM as context to generate an answer.
    :param user_input: This is the natural language question that is passed in through the app.
    py file.
    :return: The answer to your question from the LLM based on the context that was provided by
    the KNN search of OpenSearch.
```

```
8   """
9   import json
10  # creating an embedding of the user input to perform a KNN search with
11  # userVectors = get_embedding_v2(user_input,bedrockClient)
12  # the query below is the semantic search using an embedding of the user input
13  # query_vector = {"size": 3, "query": {"knn": { vector_field : {"vector": userVectors, "k":
    3}}}, "_source": True, "fields": ["text","borarag-cybersecurity-date",'borarag-
    cybersecurity-source']}
14
15  # the query below is a keyword search using the user input
16  query_keyword = {"size":3, "query": {"match": {"text": user_input}}, "_source": True, "
    fields": ["text","borarag-cybersecurity-date",'borarag-cybersecurity-source']}
17
18  # performing the search on OpenSearch passing in the query parameters constructed above
19  response = opensearchClient.search(
20      body=query_keyword,
21      index=index
22  )
23
24  # Format Json responses into text
25  context = ""
26  source = ""
27  # iterating through all the findings of Amazon openSearch and adding them to a single
    string to pass in as context
28  for i in response["hits"]["hits"]:
29      outputtext = i["fields"]["text"]
30      documentDate = i['fields']['borarag-cybersecurity-date'][0]
31      documentSource = i['fields']['borarag-cybersecurity-source'][0]
```

```
32        source = "Source " + str(i) + "- information system source: " + documentSource  + " available in the log with date " + documentDate + " contains the following:\n"
33        context =  source + str(outputtext)
34
35    #     similaritysearchResponse = similaritysearchResponse
36     # The prompt for Anthropic follows the following format: \n\nHuman: \n Task context \n Tone context \n background data and documents \n \
37        # Detailed task description and rules \n examples \n conversation history \n immediate task description or request \n thinking step by step \n \
38            # output formatting \n\nAssistant: \n\n
39
40    prompt_data = f"""\n\nHuman: You will be acting as an AI assistant called Bora-bot, created by the company Bora Pharmaceuticals.
41    Your goal is to answer questions about Bora cybersecurity, based on the logs from the vulnerability scanning OpenVAS, the end-point protection
42    CrowdStrike, the Active Directory environment for authentication from Microsoft Graph, and the browsing logs for each user in the company.
43    The logs refer to users, computers, browsing activity, directory events, sign-in events, vulnerability scans and related elements.
44    You will be replying to users on a chat interface. Respond to the question always in the character of Bora-bot.
45
46    You are a friendly, knowledgeable and helpful AI assistant.
47
48    Use the following information as Context: {context}
49
50    Here are some important rules for the interaction:
51    - Always stay in character, as Bora-bot, an AI assistant expert on cybersecurity for Bora Pharmaceuticals.
```

52 *– If the user, computer, or other elements can't be found in the logs, say "Sorry, can't find any relevant information in the logs"*

53 *– If someone asks something irrelevant, say "Sorry, as an AI assistant, do you have any question about Bora's cybersecurity?"*

54 *– Do not mention Anthropic in your responses.*

55 *– Answer in detail with no preamble. Do not repeat text in your answer unless it is necessary.*

56 *– If the question is complex, break it down into smaller questions and answer each one separately. Think step by step.*

57 *– Do not make answers up. If you don't know the answer, say "Sorry, I don't know the answer to that question."*

58

59 *Here is an example of a conversation:*

60 *<example>*

61 *Human: Hi, what can you tell me about user Miranda Kee?*

62 *Assistant: The user Miranda Kee appears in the following information system logs:*

63 *– According to Active Directory, the user was created on 2020−11−20, with the title of "Production compliance specialist" in the Production department.*

64 *– She last changed her password on 2023−12−26.*

65 *– According to the sign−in records, the user last login was on 2024−01−15 using the computer CAWL−74059, located in Toronto, Ontario, Canada. She is MFA enabled.*

66 *– According to browsing logs, the user navigated to websites such as www.sail.ca and www.mountainwarehouse.ca on 2022−06−25.*

67 *</example>*

68

69 *Here is another example about a computer:*

70 *<example>*

71 *Human: Hi, what can you tell me about computer CAWL−90667 and user Jake Smith?*

```
72  Assistant: Sorry, could not find a computer with that name, nor a user with that name in
    the logs.
73  </example>
74
75  Here is the user's question: <question> {user_input} </question>
76
77  Format your answer in a way that is easy to read and understand, using bullet points if
    necessary.
78  If you found relevant information from the logs, include it in the answer.
79  Never make up information. If you don't know the answer, say "Sorry, I don't know the
    answer to that question."
80
81  \n\nAssistant:
82
83  """
84  # Configuring the model parameters, preparing for inference
85  body = json.dumps({"prompt": prompt_data,
86                     "max_tokens_to_sample": 4096,
87                     "temperature": 0,
88                     "top_k": 250,
89                     "top_p": 0.1,
90                     "stop_sequences": []
91                     })
92
93  # Run infernce on the LLM
94  # Configuring the specific model you are using
95  modelId = "anthropic.claude-instant-v1"  # change this to use a different version from the
    model provider
96  accept = "application/json"
```

```
97      contentType = "application/json"
98      # invoking the bedrock API, passing in all specific parameters
99      response = bedrockClient.invoke_model(body=body,
100                                            modelId=modelId,
101                                            accept=accept,
102                                            contentType=contentType)
103
104     # loading in the response from bedrock
105     response_body = json.loads(response.get('body').read())
106     # retrieving the specific completion field, where you answer will be
107     answer = response_body.get('completion')
108     # returning the answer as a final result, which ultimately gets returned to the end user
109     return answer
```

Explanation The final stage in the process is the user interface with the LLM, enabling the chat capabilities that explain the data collected.

```
1   """
2  This program:
3  - Instantiates the Bedrock client and the OpenSearch collection (thus, defining what data source
       is used for the chat).
4  - Passes those values to the answer_query_v2 library, which performs the context search and
       passes to LLM
5  - Gets the answer from LLM and formats it as a chat
6  """
7  import os, json
8  import streamlit as st
9  from BoraRAG_core_functions import answer_query_v4_cybersecurity, create_clients_v2,
       answer_query_v3_controlled_documents
```

```
10 from dotenv import load_dotenv
11
12 load_dotenv()
13 #bedrockClient, opensearchClient = create_clients_v2(os.getenv('AWS_ACCESS_KEY_ID'), os.
       getenv('AWS_SECRET_ACCESS_KEY'),os.getenv('opensearch_host_RAGControlledDocs'))
14 bedrockClient, opensearchClient = create_clients_v2(os.getenv('AWS_ACCESS_KEY_ID'), os.
       getenv('AWS_SECRET_ACCESS_KEY'),os.getenv('opensearch_host_RAGCybersecurity'))
15 # Header/Title of streamlit app
16
17 st.title(f"""Bora-bot: your AI assistant""")
18 # configuring values for session state
19 if "messages" not in st.session_state:
20     st.session_state.messages = []
21 # writing the message that is stored in session state
22 for message in st.session_state.messages:
23     with st.chat_message(message["role"]):
24         st.markdown(message["content"])
25
26 # evaluating st.chat_input and determining if a question has been input
27 if question := st.chat_input("Ask Bora-bot your question"):
28     # with the user icon, write the question to the front end
29     with st.chat_message("user"):
30         st.markdown(question)
31     # append the question and the role (user) as a message to the session state
32     st.session_state.messages.append({"role": "user",
33                                       "content": question})
34
35     # respond as the assistant with the answer
36     with st.chat_message("assistant"):
```

```
37          # making sure there are no messages present when generating the answer
38          message_placeholder = st.empty()
39          # putting a spinning icon to show that the query is in progress
40          with st.status("Determining the best possible answer!", expanded=True) as status:
41              # passing the question into the OpenSearch search function, which later invokes the
        llm
42
43              #answer = answer_query_v3_controlled_documents(question,bedrockClient,
        opensearchClient,os.getenv('vector_index_name_RAGControlledDocs'),os.getenv('
        vector_field_name_RAGControlledDocs'))
44              answer = answer_query_v4_cybersecurity(question,bedrockClient,opensearchClient,os.
        getenv('vector_index_name_RAGCybersecurity_v2'),os.getenv('
        vector_field_name_RAGCybersecurity'))
45              # writing the answer to the front end
46              message_placeholder.markdown(f"{answer}")
47              # showing a completion message to the front end
48              status.update(label="Question Answered...", state="complete", expanded=False)
49      # appending the results to the session state
50      st.session_state.messages.append({"role": "assistant",
51                                        "content": answer})
```

Bibliography

[1] Alan R. Hevner et al. “Design Science in Information Systems Research”. In: (Mar. 2004), p. 32 (cit. on pp. 12, 17–20, 105, 111, 115, 116).

[2] Alan Turing. “Computing Machinery and Intelligence”. In: *Mind* (1950), pp. 433–60. (Visited on 02/16/2024) (cit. on p. 1).

[3] Adel Alshamrani et al. “A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities”. In: *IEEE Communications Surveys & Tutorials* 21.2 (2019), pp. 1851–1877. ISSN: 1553-877X, 2373-745X. DOI: 10.1109/COMST.2019.2891891. (Visited on 10/29/2023) (cit. on p. 24).

[4] James M. Anderson. “Why We Need a New Definition of Information Security”. In: *Computers and Security* 22.4 (2003), pp. 308–313. ISSN: 01674048. DOI: 10.1016/S0167-4048(03)00407-3 (cit. on p. 33).

[5] Andrey Kurenkov. “A ’Brief’ History of Neural Nets and Deep Learning”. In: (2016) (cit. on pp. 1, 16).

[6] Alejandro Barredo Arrieta et al. “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI”. In: *Information Fusion* 58 (June 2020), pp. 82–115. ISSN: 15662535. DOI: 10.

1016/j.inffus.2019.12.012. (Visited on 08/08/2020) (cit. on pp. 56, 57, 61, 63).

[7] Nauman Bin Ali et al. "What Is DevOps?: A Systematic Mapping Study on Definitions and Practices What Is DevOps? A Systematic Mapping Study on Definitions and Practices". In: 2016. DOI: 10.1145/2962695.2962707. (Visited on 08/14/2018) (cit. on p. 70).

[8] Rishi Bommasani et al. *On the Opportunities and Risks of Foundation Models*. July 2022. arXiv: 2108.07258 [cs]. (Visited on 11/05/2023) (cit. on p. 52).

[9] Tom B. Brown et al. "Language Models Are Few-Shot Learners". In: *arXiv:2005.14165 [cs]* (July 2020). arXiv: 2005.14165 [cs]. (Visited on 05/14/2021) (cit. on pp. 34, 101).

[10] Davide Castelvecchi. "Can We Open the Black Box of AI?" In: *Nature News* 538.7623 (Oct. 2016), p. 20. DOI: 10.1038/538020a. (Visited on 08/21/2020) (cit. on p. 54).

[11] Jan Chorowski et al. "Attention-Based Models for Speech Recognition". In: *arXiv:1506.07503 [cs, stat]* (June 2015). arXiv: 1506.07503 [cs, stat]. (Visited on 07/15/2021) (cit. on p. 45).

[12] Junyoung Chung et al. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *arXiv:1412.3555 [cs]* (Dec. 2014). arXiv: 1412.3555 [cs]. (Visited on 02/26/2020) (cit. on p. 41).

[13] Courtenay Cotton and Biran, Or. "Explanation and Justification in Machine Learning: A Survey". In: (2017), p. 6 (cit. on pp. 54, 56).

[14] J W Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 2014. ISBN: 978-1-4522-7461-4. DOI: 10.1007/s13398-014-0173-7.2. arXiv: 1011.1669v3 (cit. on p. 12).

[15] John W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 4th ed. Thousand Oaks: SAGE Publications, 2014. ISBN: 978-1-4522-2609-5 978-1-4522-2610-1 (cit. on p. 12).

[16] CrowdStrike. *Global 2023 Threat Report*. Tech. rep. 2023 (cit. on pp. 25, 26).

[17] William H. DeLone and Ephraim R. McLean. "Information Systems Success: The Quest for the Dependent Variable". In: *Information Systems Research* 3.1 (1992), pp. 60–95. ISSN: 1047-7047. JSTOR: 23010781. (Visited on 02/09/2022) (cit. on pp. 94, 95, 100).

[18] William H. DeLone and Ephraim R. McLean. "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update". In: *Journal of Management Information Systems* 19.4 (2003), pp. 9–30. ISSN: 07421222. DOI: 10.1080/07421222.2003.11045748 (cit. on pp. 94, 95, 102).

[19] James DeLuccia et al. *DevOps Audit Defense Toolkit*. Tech. rep. 2015 (cit. on p. 33).

[20] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv:1810.04805 [cs]* (May 2019). arXiv: 1810.04805 [cs]. (Visited on 11/26/2020) (cit. on p. 50).

[21] Georg Disterer. "ISO/IEC 27000, 27001 and 27002 for Information Security Management". In: *Journal of Information Security* 04.02 (2013), pp. 92–100. ISSN: 2153-1234. DOI: 10.4236/jis.2013.42011 (cit. on p. 77).

[22] Pedro Domingos. "A Few Useful Things to Know about Machine Learning". In: *Communications of the ACM* 55.10 (2012), p. 78. ISSN: 00010782. DOI: 10.1145/2347736.2347755. arXiv: cs/9605103 (cit. on p. 58).

[23] Guozhu Dong and Huan Liu. "Feature Engineering for Machine Learning and Data Analytics". In: c (2018) (cit. on p. 58).

[24] Robert Dubin. "Theory Building in Applied Areas". In: 1978 (cit. on p. 11).

[25] Andrej Dyck et al. "Towards Definitions for Release Engineering and DevOps". In: *Proceedings - 3rd International Workshop on Release Engineering, RELENG 2015*. 2015, p. 3. ISBN: 978-1-4799-1934-5. DOI: 10.1109/RELENG.2015.10. (Visited on 06/05/2018) (cit. on p. 70).

[26] Ethem Alpaydin. *Introduction to Machine Learning*. Third. 2014 (cit. on p. 36).

[27] Stefan Fenz and Andreas Ekelhart. "Formalizing Information Security Knowledge". In: *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security - ASIACCS '09*. ACM, 2009, p. 183. ISBN: 978-1-60558-394-5. DOI: 10.1145/1533057.1533084 (cit. on p. 32).

[28] Jose Fumo. *Types of Machine Learning Algorithms You Should Know*. https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861. Aug. 2017. (Visited on 02/28/2024) (cit. on p. 37).

[29] Dennis A. Gioia and Evelyn Pitre. "Multiparadigm Perspectives on Theory Building". In: *Academy of Managment Review* 15.4 (1990), pp. 584–602. ISSN: 0363-7425. DOI: 10.5465/AMR.1990.4310758 (cit. on p. 13).

[30] *Global Cybersecurity Outlook 2023*. https://initiatives.weforum.org/global-cyber-outlook/home. (Visited on 10/22/2023) (cit. on p. 19).

[31] Ian Goodfellow et al. "Deep Learning". In: *Deep Learning* December (2016), p. 785. ISSN: 1432122X. DOI: 10.1016/B978-0-12-391420-0.09987-X. arXiv: 1011.1669v3 (cit. on pp. 16, 39).

[32] *GPT-4 Technical Report*. Tech. rep. (Visited on 11/05/2023) (cit. on p. 51).

[33] Shirley Gregor. "Design Theory in Information Systems". In: *Australasian Journal of Information Systems* 10.1 (Nov. 2002). ISSN: 1449-8618, 1449-8618. DOI: 10.3127/ajis.v10i1.439. (Visited on 07/12/2020) (cit. on p. 17).

[34] Shirley Gregor. "The Nature of Theory in Information Systems". In: *MIS Quaterly* 30.3 (2006), pp. 611–642. ISSN: 0276-7783. DOI: 10.1080/02683960220000177 25. JSTOR: 25148742 (cit. on pp. 5, 11).

[35] Shirley Gregor and Alan R. Hevner. "Positioning and Presenting Design Science Research for Maximum Impact". In: *MIS Quarterly* 37.2 (Feb. 2013), pp. 337–355. ISSN: 02767783, 21629730. DOI: 10.25300/MISQ/2013/37.2.01. (Visited on 12/04/2021) (cit. on pp. 14, 18, 20, 126).

[36] Shirley Gregor and David Jones. "THE ANATOMY OF A DESIGN THEORY". In: (), p. 60 (cit. on p. 12).

[37] Shirley Gregor and David Jones. "The Anatomy of a Design Theory". In: (2007), p. 60 (cit. on p. 11).

[38] David Gunning. *Explainable Artificial Intelligence (XAI)*. 2017 (cit. on p. 55).

[39] Moritz Hardt and Benjamin Recht. *Patterns, Predictions, and Actions*. Oct. 2021 (cit. on p. 53).

[40] Shon Harris. *All in One Cissp*. 2008. ISBN: 978-0-07-149786-2 (cit. on p. 32).

[41] Alan Hevner and Samir Chatterjee. *Design Research in Information Systems: Theory and Practice*. Vol. 22. Integrated Series in Information Systems. Boston, MA: Springer US, 2010. ISBN: 978-1-4419-5652-1 978-1-4419-5653-8. DOI: 10.1007/978-1-4419-5653-8. (Visited on 02/11/2024) (cit. on pp. 17, 21).

[42] Alan R Hevner et al. "Design Science in Information Systems Research". In: (2004), p. 33 (cit. on pp. 12, 14, 15, 19).

[43] High-level expert group on AI (AI HLEG). *Ethics Guidelines for Trustworthy AI*. Tech. rep. 2019 (cit. on pp. 55, 56).

[44] Geoffrey Hinton et al. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6 (Nov. 2012), pp. 82–97. ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2205597. (Visited on 11/05/2023) (cit. on p. 51).

[45] *How Did Supercomputer Watson Beat Jeopardy Champion Ken Jennings? — TED Blog*. (Visited on 01/30/2022) (cit. on pp. 2, 34).

[46] Chris Huxham and Siv Vangen. "Researching Organizational Practice Through Action Research: Case Studies and Design Choices". In: *Organizational Research Methods* 6.3 (2003), pp. 383–403. ISSN: 10944281. DOI: 10.1177/1094428103254454 (cit. on p. 17).

[47] *IBM100 - Deep Blue*. http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/. CTB14. Mar. 2012. (Visited on 08/16/2020) (cit. on pp. 2, 34).

[48] Juhani Iivari. "A Paradigmatic Analysis of Information Systems As a Design Science". In: (2007). (Visited on 12/05/2021) (cit. on pp. 10, 12).

[49] Ilona Ilvonen et al. "Towards a Business-Driven Process Model for Knowledge Security Risk Management: Making Sense of Knowledge Risks". In: *International Journal of Knowledge Management* 11.4 (2015), p. 18. ISSN: 15480658. DOI: 10.4018/IJKM.2015100101 (cit. on p. 33).

[50] Information Systems Audit and Control Association. *COBIT 2019 Design Guide Designing an Information and Technology Governance Solution*. 2019. ISBN: 978-1-60420-765-1 978-1-60420-761-3 (cit. on pp. 31, 32).

[51] Information Systems Audit and Control Association. *COBIT 2019 Framework Governance and Management Objectives*. 2019. ISBN: 978-1-60420-764-4 (cit. on pp. 29, 30, 54, 75).

[52] Information Systems Audit and Control Association. *COBIT 2019 Framework: Introduction and Methodology*. 2012. ISBN: 978-1-60420-763-7. (Visited on 11/25/2021) (cit. on pp. 27, 30, 31, 68).

[53] Project Management Institute. *A Guide to the Project Management Body of Knowledge*. Seventh edition. Pennsylvania: Project Management Institute, Aug. 2021. ISBN: 978-1-62825-664-2 (cit. on p. 120).

[54] International Federation of Accountants. "Enterprise Governance: Getting the Balance Right". In: (2004), p. 8 (cit. on p. 28).

[55] *Introducing ChatGPT*. https://openai.com/blog/chatgpt. (Visited on 11/05/2023) (cit. on pp. 10, 51).

[56] *ISO Guide 73 - Risk Management Vocabulary*. 2009 (cit. on p. 32).

[57] *Just Ask for Generalization*. https://evjang.com/2021/10/23/generalization.html. Oct. 2021. (Visited on 02/18/2022) (cit. on p. 53).

[58] Andrej Karpathy et al. "Visualizing and Understanding Recurrent Networks". In: *arXiv:1506.02078 [cs]* (Nov. 2015). arXiv: 1506.02078 [cs]. (Visited on 08/28/2020) (cit. on p. 61).

[59] Gene Kim et al. *The DevOps Handbook*. 2016 (cit. on p. 70).

[60] Gene Kim et al. *The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win*. 2013. ISBN: 978-0-9882625-9-1 (cit. on p. 70).

[61] John P. Kotter. *Leading Change*. Boston, Mass: Harvard Business Review Press, 2012. ISBN: 978-1-4221-8643-5 (cit. on p. 28).

[62] Yann LeCun et al. "Deep Learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. (Visited on 09/04/2020) (cit. on pp. 1, 38).

[63] Yann Lecun et al. "Deep Learning". In: *Nature* 521.7553 (2015), pp. 436–444. ISSN: 14764687. DOI: 10.1038/nature14539. arXiv: 1312.6184v5 (cit. on pp. 16, 39).

[64] Allen S. Lee. "Integrating Positivist and Interpretive Approaches to Organizational Research". In: *Organization science* 2.4 (1991), pp. 342–365 (cit. on p. 4).

[65] Jiwei Li et al. "Visualizing and Understanding Neural Models in NLP". In: *arXiv:1506.01066 [cs]* (Jan. 2016). arXiv: 1506.01066 [cs]. (Visited on 08/28/2020) (cit. on p. 61).

[66] F. T. Liu et al. "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. Dec. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17 (cit. on p. 90).

[67] Steve Lohr. "IBM Is Selling off Watson Health to a Private Equity Firm." In: *The New York Times* (Jan. 2022). ISSN: 0362-4331. (Visited on 10/21/2023) (cit. on p. 3).

[68] Steve Lohr. "What Ever Happened to IBM's Watson?" In: *The New York Times* (July 2021). ISSN: 0362-4331. (Visited on 10/21/2023) (cit. on p. 3).

[69] Kevin Lu et al. "Pretrained Transformers as Universal Computation Engines". In: *arXiv:2103.05247 [cs]* (June 2021). arXiv: 2103.05247 [cs]. (Visited on 02/18/2022) (cit. on p. 53).

[70] Mohammed Lubbad. *The Ultimate Guide to GPT-4 Parameters: Everything You Need to Know about NLP's Game-Changer*. Oct. 2023. (Visited on 10/21/2023) (cit. on p. 2).

[71] *Machine Learning Algorithms*. Aug. 2023. (Visited on 02/28/2024) (cit. on p. 37).

[72] David E Mann and Steven M Christey. "Towards a Common Enumeration of Vulnerabilities". In: () (cit. on p. 26).

[73] J.A. Maxwell. "Qualitative Research Design : An Interactive Approach". In: *Qualitative research design : an interactive approach* (2013), p. 201 (cit. on p. 17).

[74] John McCarthy. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence". In: (1955), p. 3 (cit. on p. 33).

[75] Pamela McCorduck. *Machines Who Think*. Vol. 5. 2004. ISBN: 1-56881-205-1 (cit. on p. 34).

[76] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and Their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 3111–3119. (Visited on 05/04/2020) (cit. on p. 41).

[77] Robert R. Moeller. *Executive's Guide to COSO Internal Controls: Understanding and Implementing the New Framework*. Wiley Corporate F&A Series. Hoboken, New Jersey: John Wiley and Sons, Inc, 2014. ISBN: 978-1-118-62641-2 (cit. on p. 32).

[78] *Neural Networks: Feedforward and Backpropagation Explained*. https://mlfromscratch.com/neural-networks-explained/. Aug. 2019. (Visited on 04/06/2020) (cit. on p. 38).

[79] Long Ouyang et al. *Training Language Models to Follow Instructions with Human Feedback*. Mar. 2022. arXiv: 2203.02155 [cs]. (Visited on 11/05/2023) (cit. on p. 51).

[80] Anne E Pezalla et al. "Researching the Researcher-as-Instrument: An Exercise in Interviewer Self-Reflexivity". In: *Qualitative Research* 12.2 (Apr. 2012), pp. 165–185. ISSN: 1468-7941. DOI: 10.1177/1468794111422107. (Visited on 02/17/2024) (cit. on p. 12).

[81] Michael E. Porter. "Strategy and the Internet". In: *Harvard Business Review* (Mar. 2001). ISSN: 0017-8012. (Visited on 02/16/2022) (cit. on p. 69).

[82] Michael E. Porter and James E. Heppelmann. "How Smart, Connected Products Are Transforming Competition". In: *Harvard Business Review* (Nov. 2014). ISSN: 0017-8012. (Visited on 02/16/2022) (cit. on p. 69).

[83] Michael E. Porter and Victor E. Millar. "How Information Gives You Competitive Advantage". In: *Harvard Business Review* (July 1985). ISSN: 0017-8012. (Visited on 02/16/2022) (cit. on p. 69).

[84] Samira Pouyanfar et al. "A Survey on Deep Learning: Algorithms, Techniques, and Applications". In: *ACM Computing Surveys* 51.5 (Sept. 2019), pp. 1–36. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3234150. (Visited on 02/05/2022) (cit. on p. 88).

[85] Alec Radford et al. "Improving Language Understanding by Generative Pre-Training". In: () (cit. on p. 51).

[86] Raymond Perrault et al. *Artificial Intelligence Index - 2019 Annual Report*. 2019. (Visited on 05/04/2020) (cit. on p. 34).

[87] Center for Drug Evaluation and Research. "Facts About the Current Good Manufacturing Practice (CGMP)". In: *FDA* (Fri, 02/16/2024 - 14:27). (Visited on 02/17/2024) (cit. on p. 30).

[88] Marco Tulio Ribeiro et al. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, Aug. 2016, pp. 1135–1144. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939778. (Visited on 08/30/2020) (cit. on pp. 61, 62).

[89] Stuart Russell and Peter Norvig. *AI Book*. 2009. ISBN: 978-0-13-604259-4. DOI: 10.1017/S0269888900007724. arXiv: 1011.1669v3 (cit. on p. 2).

[90] John C. Scott and Barney G. Glaser. *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Vol. 36. 2006. ISBN: 0-202-30260-1. DOI: 10.2307/2094063. arXiv: gr-qc/9809069v1 (cit. on p. 4).

[91] Sepp Hochreiter and Jurgen Schmidhuber. *Long Short-Term Memory*. 1997 (cit. on p. 42).

[92] Herbert Alexander Simon. *The Sciences of the Artificial*. 3. ed., [Nachdr.] Cambridge, Mass.: MIT Press, 2008. ISBN: 978-0-262-19374-0 978-0-262-69191-8 (cit. on pp. 14, 34).

[93] Stanford University. *AI Index Report 2022*. (Visited on 02/17/2024) (cit. on p. 35).

[94] Stanford University. *AI Index Report 2023*. 2023. (Visited on 02/17/2024) (cit. on pp. 35, 121).

[95] Adrijana Biba Starman. "The Case Study as a Type of Qualitative Research". In: (), p. 17 (cit. on p. 13).

[96] Hendrik Strobelt et al. "LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks". In: *IEEE Transactions on Visualization and Computer Graphics* 24.1 (Jan. 2018), pp. 667–676. ISSN: 1941-0506. DOI: 10.1109/TVCG.2017.2744158 (cit. on p. 61).

[97] Robert I Sutton and Barry M Staw. "What Theory Is Not". In: *Administrative Science Quarterly (ASQ)* 40.3 (1995), pp. 371–384. ISSN: 00018392. DOI: 10.2307/2393788. arXiv: 1011.1669v3. (Visited on 07/24/2018) (cit. on p. 13).

[98] *The Triple Constraint*. https://www.pmi.org/learning/library/triple-constraint-erroneous-useless-value-8024. (Visited on 02/19/2024) (cit. on p. 120).

[99] *To Understand Language Is to Understand Generalization*. https://evjang.com/2021/12/17/lang-generalization.html. Dec. 2021. (Visited on 02/18/2022) (cit. on p. 53).

[100] *TOGAF — Www.Opengroup.Org*. https://www.opengroup.org/togaf. (Visited on 02/10/2024) (cit. on p. 68).

[101] Akond Ashfaque Ur Rahman and Laurie Williams. "Security Practices in DevOps". In: *Proceedings of the Symposium and Bootcamp on the Science of Security - HotSos '16*. Pittsburgh, Pennsylvania: ACM Press, 2016, pp. 109–111. ISBN: 978-1-4503-4277-3. DOI: 10.1145/2898375.2898383. (Visited on 10/15/2018) (cit. on p. 70).

[102] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008. (Visited on 10/01/2020) (cit. on pp. 45, 46, 49).

[103] Isabelle Walsh et al. "What Grounded Theory Is... A Critically Reflective Conversation Among Scholars". In: *Organizational Research Methods* 18.4 (2015), pp. 581–599. ISSN: 15527425. DOI: 10.1177/1094428114565028 (cit. on p. 4).

[104] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. Jan. 2023. arXiv: 2201.11903 [cs]. (Visited on 02/11/2024) (cit. on p. 101).

[105] *What Is Microservices Architecture?* https://cloud.google.com/learn/what-is-microservices-architecture. (Visited on 02/19/2024) (cit. on p. 123).

[106] ME Whitman. "Enemy at the Gate: Threats to Information Security". In: *Communications of the ACM* 46.8 (2003), pp. 91–95. ISSN: 0001-0782. DOI: 10.1145/859670.859675 (cit. on p. 32).

[107] Norman Wilde et al. "Security for Devops Deployment Processes: Defenses, Risks, Research Directions". In: *International Journal of Software Engineering & Applications (IJSEA)* 7.6 (2016), p. 16. DOI: 10.5121/ijsea.2016.7601 (cit. on p. 70).

[108] *Written Policy on Electronic Monitoring of Employees — Your Guide to the Employment Standards Act — Ontario.Ca.* http://www.ontario.ca/document/your-guide-employment-standards-act-0/written-policy-electronic-monitoring-employees. (Visited on 02/17/2024) (cit. on p. 76).

[109] Yoshua Bengio, Patrice Simard, Paolo Frasconi. "Learning Long-Term Dependencies with Gradient Descent Is Difficult". In: (Mar. 1994). (Visited on 02/26/2020) (cit. on p. 39).

[110] Zachary C. Lipto. "The Mythos of Model Interpretability". In: (2016) (cit. on pp. 54, 57, 61).