CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Evaluating Retrieval-Augmented Generation for Automated Regulatory Gap Analysis in the Information Security domain

Master's Thesis in Computer science and engineering

(Jenifa Mariyanayagam)

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2026

Master's Thesis 2026

Evaluating Retrieval-Augmented Generation for Automated Regulatory Gap Analysis in the Information Security domain

Jenifa Mariyanayagam

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2026

Evaluating Retrieval-Augmented Generation for Automated Regulatory Gap Analysis in the Information Security domain
Jenifa Mariyanayagam

Supervisor: Farnaz Fotrousi, Department of Data and Information Technology
Examiner: Jennifer Horkoff, Department of Data and Information Technology

Master's Thesis 2026
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Acknowledgments, dedications, and similar personal statements in this thesis reflect the author's own views.

Typeset in LaTeX
Gothenburg, Sweden 2026

Evaluating Retrieval-Augmented Generation for Automated Regulatory Gap Analysis in the Information Security domain
Jenifa Mariyanayagam
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Regulatory compliance in information security requires organisations to map their controls and mitigations against frameworks such as ISO/IEC 27001 and the NIS2 Directive. This process is traditionally manual, resource-intensive, and difficult to scale. Large Language Models (LLMs) offer potential for automating compliance tasks, but their tendency to hallucinate and produce unverifiable outputs limits their applicability in contexts where findings must be defensible and auditable.
This thesis investigates whether Retrieval-Augmented Generation (RAG) improves the accuracy, recommendation quality, and clause-level traceability of LLM-assisted regulatory gap analysis compared to a non-grounded baseline. A controlled computational experiment was conducted following Wohlin et al.'s experimentation framework. Two large language models (GPT-5.1 and Claude Opus 4.6) were evaluated under two configurations (baseline and RAG) on a dataset of 93 compliance risks mapped against ISO/IEC 27002 and ENISA NIS2 Technical Implementation Guidance. Outputs were evaluated against an expert-validated Golden Standard Reference using clause prediction accuracy, coverage assessment accuracy, mitigation quality metrics, and LLM-as-Judge evaluation.
Results show that RAG significantly improves gap detection for GPT-5.1, increasing ISO clause prediction accuracy from 43.0% to 79.6% ($p = 0.000001$) and enabling ENISA clause identification that is impossible from parametric knowledge alone (0% to 33.3%). Coverage assessment accuracy improved from 78.5% to 98.9% under relaxed evaluation. For Claude Opus 4.6, which already achieves 86.0% ISO accuracy from parametric knowledge, RAG does not significantly improve clause prediction but remains essential for ENISA identification. RAG acts as an equaliser across model architectures, reducing the inter-model performance gap from 43 percentage points to 5.3. LLM-as-Judge evaluation shows RAG produces higher-quality mitigations (winning 80.7% of pairwise comparisons), while expert validation confirms both pipelines achieve high ISO mitigation quality.
The study contributes empirical evidence on the use of retrieval-augmented grounding for information security GRC gap analysis, demonstrating that RAG improves accuracy and enables traceability for regulatory frameworks absent from LLM pretraining data.

Keywords: retrieval-augmented generation, regulatory compliance, gap analysis, large language models, information security, traceability, ISO 27001, NIS2, grounding, evaluation

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Farnaz Fotrousi, and my examiner, Jennifer Horkoff, for their invaluable guidance throughout this thesis. Their willingness to respond to questions, provide feedback, and offer support even on the most fundamental concerns made this work possible. I am deeply grateful for their consistent availability and encouragement, which kept me motivated and on track from start to finish.
I would also like to thank the AI and GRC teams at Stratsys for providing the technical guidance, domain expertise, and empirical context that grounded this research in real-world practice.

Thanks,
(Jenifa Mariyanayagam), Gothenburg, June 2026

Contents

List of Figures

List of Tables

1

Introduction

Governance, Risk Management, and Compliance (GRC) serves as a framework that helps organizations align their operations, risk profile, and regulatory obligations with their overall goals. GRC processes ensure continuous adherence to legal and ethical standards and help organizations avoid legal penalties, fines, and reputational harm [1]. Organizations apply the principles of GRC across multiple domains, including Information Security. These principles dictate how critical data assets are protected against an escalating landscape of cyber threats, requiring that organizational artifacts—such as risk descriptions, mitigations, and controls—be continuously mapped against stringent regulatory frameworks like ENISA (NIS2) and ISO/IEC 27001 [8, 9, 10].

Traditionally, regulatory compliance is established through internal and external audits, during which an organization's documented artifacts (policies, control descriptions, and evidence records) are validated against regulatory requirements [2]. When this validation fails, it indicates compliance gaps, such as missing coverage, incomplete policies, or claims that cannot be verified during an audit.

In practice, regulatory gap analysis is difficult because regulations are often long and complex, with many clauses requiring mapping against various documents and processes within an organization. Although organizations have adopted modern systems and web-based platforms to document and maintain their processes and controls, compliance analysis is predominantly performed manually by human experts [11]. This makes the process resource-intensive, time-consuming, and difficult to scale. It can also lead to inconsistent results, particularly when compliance claims must be justified across large volumes of documentation [12]. However, due to the critical nature of the regulations and the gravity of consequences, compliance gap findings must be complete and defensible [2].

To address the challenges with scaling, recent research has explored the use of Large Language Models (LLMs) to support text-intensive compliance tasks and automation of regulatory analysis. However, current literature highlights challenges with the reliability of LLM responses, which manifest as hallucination or false positives [13, 14]. Moreover, the general 'black-box' behaviour of LLMs poses a hurdle for industry-grade automation solutions, where transparency and explainability are important [15].

This vulnerability to hallucination has threatened the application of LLMs in various sensitive domains beyond GRC. Recent research exploring the mitigation of hallucination in LLMs has successfully demonstrated the use of techniques such as Retrieval-Augmented Generation (RAG) [14, 17, 38].

There is a lack of academic work that applies hallucination mitigation in the context of GRC workflows, specifically in the Information security domain. In particular, no prior study empirically evaluates how RAG can affect the accuracy and traceability of LLM-assisted automation of regulatory analysis.

This study proposes to bridge this gap by empirically evaluating retrieval-augmented grounding in terms of gap-detection accuracy and completeness, while also assessing auditability through traceability. Within this framework, traceability implies verifying whether each identified gap or suggested mitigation is mapped directly to the relevant regulatory clause, as well as supporting organizational artifacts. The ultimate aim of this study is to provide practical evidence on how grounding can be used to support regulatory gap analysis, offering a compliance system that is scalable, cost-effective, and auditable—without the need for expensive model fine-tuning.

This thesis investigates the question: *"Can LLM-based tools automate compliance gap analysis and produce accurate, defensible results?"*

To explore this, the thesis addresses the following research questions:

- **RQ1:** To what extent does retrieval-augmented generation affect the precision, recall, and traceability of LLM-assisted regulatory *gap detection* (clause identification and coverage assessment), compared to a non-grounded baseline?
- **RQ2:** To what extent does the choice of LLM affect the performance of automated compliance gap detection?
- **RQ3:** To what extent does retrieval-augmented generation affect the quality of generated *remediation recommendations* (improved risk, mitigation, and control statements), compared to a non-grounded baseline?

To answer these questions, the study follows a computational experimental approach [27]. A baseline LLM-based pipeline is compared with a retrieval-augmented configuration applied to the same regulatory and organisational data. The LLM outputs are evaluated against the Golden Standard Reference (GSR), enabling the assessment of both performance and traceability.

The contribution of this thesis is to provide empirical evidence on the use of retrieval-augmented grounding for information security GRC gap analysis—specifically operational security controls and incident management under ENISA (NIS2) and ISO/IEC 27001—evaluating its impact on accuracy and traceability compared to a non-grounded baseline.

2

Background

This chapter provides the theoretical and technical foundations for this thesis. Section 2.1 introduces the regulatory compliance frameworks used in this study. Section 2.2 discusses the application of LLMs in regulatory compliance and their limitations. Section 2.4 describes grounding techniques, including retrieval-augmented generation. Section 2.5 presents the retrieval methods employed. Section 2.6 covers prompt engineering techniques. Section 2.7 describes the evaluation approaches. Section 2.8 connects the study to requirements traceability. Section 2.9 defines the key terms used throughout this thesis.

2.1 Regulatory Compliance and Frameworks

Regulatory compliance refers to an organisation's adherence to the laws, regulations, and industry standards that apply to its operations [1]. To meet these requirements, organisations establish policies, implement controls and keep documentation that shows they are meeting these requirements.

Compliance is verified through internal and external audits. During these audits, organisational artifacts such as policies, control descriptions and evidence records are reviewed and compared against the relevant requirements [2]. If these artifacts do not satisfy the requirements, compliance gaps may arise, including missed controls, incomplete policies or claims that cannot be verified during an audit.

This study focuses on two commonly used compliance frameworks that regulate how organisations design, implement, and document their security controls and processes: the NIS2 Directive and ISO/IEC 27001.

2.1.1 ISO/IEC 27001 and ISO/IEC 27002

ISO/IEC 27001:2022 is a voluntary international standard that specifies requirements for establishing, implementing, maintaining and continually improving an information security management system (ISMS) [7]. The standard defines management system requirements in its main clauses (4–10), covering topics such as leadership, planning, support, operation, performance evaluation and improvement.

The security controls themselves are defined in a companion standard, ISO/IEC

27002:2022 [8], which provides a reference set of information security controls organised into four themes: organisational, people, physical, and technological. These controls are referenced in Annex A of ISO/IEC 27001 and are identified by control numbers such as A.5.1, A.8.25, and A.8.31. In this study, the ISO index used for retrieval contains controls from ISO/IEC 27002, as these represent the specific security requirements that organisations must implement and auditors must verify. The evaluation focuses on controls related to information security management and secure operations (e.g. A.5.1, A.8.25, A.8.29), consistent with the scope defined in Chapter 1.

2.1.2 NIS2 Directive and ENISA Technical Implementation Guidance

The NIS2 Directive (Directive (EU) 2022/2555) is a mandatory European Union legislative framework that establishes a high common level of cybersecurity across the Union [9]. It replaces the original NIS Directive and expands the scope to cover a broader range of essential and important entities, imposing obligations related to risk management, incident reporting, and supply chain security.

To support implementation of the NIS2 Directive, the European Union Agency for Cybersecurity (ENISA) published the Technical Implementation Guidance (June 2025, version 1.0) [10]. This document provides detailed technical requirements and evidence expectations organised into 13 chapters covering topics such as security policies (Chapter 1), risk management (Chapter 2), incident management (Chapter 3), business continuity (Chapter 4), supply chain security (Chapter 5), and technical measures including network security, access control, cryptography, secure development, vulnerability handling, and asset management (Chapters 6–13). Each chapter contains numbered sub-clauses (e.g. 6.6.1, 3.2.3) that specify concrete controls with accompanying guidance and examples of evidence.

In this study, the ENISA index used for retrieval contains clauses from this Technical Implementation Guidance, as these represent the specific regulatory requirements against which organisations are assessed. The evaluation covers a broad range of ENISA chapters, including clauses related to security policies, risk management, incident management, supply chain security, and technical measures such as access control, patch management, and configuration hardening.

2.2 LLMs in Regulatory Compliance

Large Language Models (LLMs) are increasingly used to support text-intensive tasks such as regulatory analysis and compliance checking. These models can process large volumes of text and assist in identifying regulatory requirements and mapping them to organisational artifacts.

However, applying LLMs in regulatory compliance introduces several challenges. Hallucination is the phenomenon where a Large Language Model (LLM) generates

outputs that are factually incorrect, unverifiable, or unsupported by the provided input or real-world evidence. These responses may include fabricated facts, incorrect citations, or logically inconsistent statements. Such inaccurate responses are classified as ungrounded [38].

Kazlaris et al. [38] categorise hallucination mitigation techniques into retrieval-based grounding, prompt engineering, and hybrid approaches, highlighting retrieval-based grounding as one of the most effective strategies.

In the context of compliance, hallucination is particularly problematic because incorrect clause references or fabricated control recommendations cannot be defended during an audit. This motivates the use of grounding techniques that anchor LLM outputs in actual regulatory text.

2.3 LLM Limitations in Long-Format Text

Wang et al. [20] find that ChatGPT's decision is sensitive to the order of labels in the prompt, and it has a clearly higher chance to select the labels at earlier positions as the answer. This phenomenon has been coined as "positional bias."

Lu et al. [21], who explored positional bias further with Mistral, Llama3, and Gemma, conclude that the underlying issue in long-context processing is a mechanistic gap where internal transformer layers successfully find the location of relevant information, but the final output layers fail to translate this internal knowledge into a correct response. They call this the "know but don't tell" phenomenon.

Du et al. [22] further confirm the issues with long contexts when using Retrieval-Augmented Generation (RAG). They also suggest a simple yet effective mitigation strategy that involves prompting the model to recite the retrieved evidence before responding.

Studies on long-context processing reveal structural limitations of LLMs that are directly relevant to regulatory gap analysis, where large volumes of text must be processed. These findings motivate the evaluation of grounding techniques in this thesis.

2.4 Grounding Techniques

To address these limitations of Large Language Models (LLMs), particularly hallucination and lack of reliability, researchers have explored techniques that ensure model outputs are based on verifiable evidence. These methods, known as grounding techniques, are designed to make the model's answers more accurate and trustworthy.

Grounding can be broadly classified into two categories:

- **Fine-tuning or post-training**, where a pre-trained model is further trained on a smaller, specialised dataset, and its internal weights are adjusted. This is a resource-intensive and time-consuming process that limits scalability.

- **Inference-time grounding**, where a model is provided with relevant context at the time of the query. Instead of altering the model's internal weights, this technique retrieves related document snippets from external databases and injects them into the prompt window.

Research on inference-time grounding techniques, such as Retrieval-Augmented Generation (RAG), shows promise in reducing hallucination [23], but it does not fully address explainability and traceability. The ability to defend and explain automated compliance assessment is equally important.

This study focuses on two inference-time grounding configurations:

Baseline (LLM-only): In this configuration, the LLM relies solely on its parametric knowledge—the information encoded in its weights during pre-training—to answer compliance queries. No external documents are retrieved. This serves as the control condition for evaluating whether retrieval augmentation improves performance.

Retrieval-Augmented Generation (RAG): RAG is a framework that enhances LLM accuracy by retrieving relevant information from external data sources and injecting it into the prompt to ground the model's responses in factual, up-to-date context [23]. Ganatra et al. [25] note that RAG, by grounding generation in external, verifiable knowledge, is key to building factual, trustworthy, and context-aware AI systems. Kozhipuram et al. [26] in their evaluation of RAG-enabled and baseline LLMs find that RAG augmentation consistently improved semantic similarity scores across multiple models.

While prior work addresses explainability and compliance automation separately, empirical studies that jointly evaluate detection accuracy and traceability in information security GRC settings remain limited. This gap motivates the experimental evaluation conducted in this thesis.

It is important to distinguish compliance checking from gap detection. *Compliance checking* determines whether an organisation satisfies a regulatory requirement (a binary pass/fail verdict). *Gap detection*, as addressed in this thesis, goes further: it identifies *which specific regulatory clauses* are not adequately covered, assesses the *degree of coverage* (not covered, partially covered, or fully covered), and generates *actionable remediation recommendations* to address identified gaps. Gap detection therefore subsumes compliance checking but additionally requires clause-level traceability and recommendation generation.

2.5 Retrieval Methods

This section provides technical background on the retrieval methods employed in the RAG pipeline. The empirical evaluation of retrieval performance (top-5 recall) is presented in Section 5.2.1.

The RAG pipeline used in this study relies on a retrieval component that searches external document indexes and returns the most relevant clauses for a given query. Three retrieval methods are commonly used:

Keyword search (BM25): BM25 (Best Matching 25) is a classical information retrieval algorithm that ranks documents based on term frequency and inverse document frequency [24]. It is effective when the query and the relevant documents share exact or near-exact vocabulary, but it cannot capture semantic similarity when different words are used to express the same concept.

Vector (semantic) search: In vector search, both queries and documents are converted into high-dimensional numerical representations (embeddings) using a neural embedding model. Retrieval is performed by computing the similarity (typically cosine similarity) between the query embedding and the document embeddings. This method captures semantic meaning, allowing it to match queries with documents that use different wording but express similar concepts. In this study, the embedding model `text-embedding-3-large` [41] is used to generate embeddings for both queries and regulatory clause chunks.

Hybrid search: Hybrid search combines both keyword-based (BM25) and vector-based retrieval to leverage the strengths of each method. The keyword component ensures that exact terminology matches are captured, while the vector component ensures that semantically related clauses are also retrieved. This study uses hybrid search as the default retrieval method, implemented via Azure AI Search [42], which combines BM25 scoring with vector similarity in a single query.

2.6 Prompt Engineering

The design of prompts plays an important role in controlling LLM behaviour and output quality. In this study, several prompt engineering techniques are applied:

Task decomposition: Instead of using a single prompt to perform the entire compliance analysis, the pipeline decomposes the task into multiple smaller subtasks, each handled by a separate LLM call [43]. This approach improves controllability, reduces ambiguity, and increases traceability of intermediate outputs.

Role prompting: Each prompt assigns a specific role to the model, such as "You are a strict classifier" or "You are a compliance auditor" [44]. Role-based instructions help constrain the model's behaviour and produce outputs consistent with the assigned persona.

Structured output prompting: All prompts in the pipeline require the model to return its response in a specific JSON format. This ensures that outputs are machine-parsable, deterministic, and directly usable in downstream pipeline steps.

These techniques are applied in combination to ensure that each LLM call in the pipeline produces focused, structured, and traceable outputs.

2.7 Evaluation Approaches

Evaluating the quality of LLM-generated compliance outputs requires both a reference standard and a scalable evaluation method.

2.7.1 Ground Truth and Reference Standards

In experimental research, a ground truth (also called a reference standard or gold standard) is a dataset of known correct answers against which system outputs are compared [27]. In this study, an expert-validated Golden Standard Reference (GSR) was created. The GSR contains 93 entries, each mapping a compliance risk to the expected regulatory clauses, mitigations, and controls from both ISO/IEC 27002 and ENISA. This serves as the reference against which the pipeline's clause selection, coverage assessment, and recommendations are evaluated.

2.7.2 LLM-as-a-Judge

Manually evaluating the quality of generated compliance recommendations across multiple configurations and samples is time-consuming and difficult to scale. Zheng et al. [29] propose the LLM-as-a-Judge approach, where a separate LLM is used to evaluate the quality of generated outputs by comparing them against a reference. This method has been shown to produce evaluations that are broadly consistent with human judgements while being significantly more scalable.

In this study, the LLM-as-a-Judge approach is used to assess precision (how much of the generated output is correct relative to the GSR) and recall (how much of the GSR's key content is captured in the generated output). Traceability is evaluated separately by checking whether the selected clause matches the expected clause(s) in the GSR.

Es et al. [28] also provide an evaluation framework (RAGAS) for assessing retrieval-augmented generation systems, which informed the design of the evaluation pipeline used in this study.

2.8 Traceability in Requirements Engineering

Regulatory gap analysis can be viewed as a requirements engineering problem. Regulatory clauses act as high-level requirements. Organisational policies and evidence records act as implementation artifacts. Determining compliance requires checking

whether appropriate trace links exist between the requirement and the supporting artifacts. This is similar to the problem of requirements traceability and trace link recovery studied in Requirements Engineering [34]. Therefore, research on automated traceability and explainable requirement analysis is directly relevant to this thesis.

Recent research has explored the use of LLMs for requirements engineering tasks, with mixed results. Seifert et al. [16] compared LLM-based defect detection with human reviewers and found that while LLMs can identify certain defects, they report fewer issues and lack thoroughness when limited to simple prompting strategies.

A systematic literature review by Cheng et al. [17] that analysed several papers (2019–May 2025) highlights that most existing work treats LLMs as "black-box" text generators. They also note that the existing literature remains largely experimental, with limited validation in industrial contexts. The review emphasises challenges related to hallucinations, traceability, and contextual reasoning, and calls for more systematic engineering approaches and evaluations using industry-grade data.

Hassine [33] used OpenAI's GPT-3.5-turbo model to recover requirements traceability links between software goal models and security requirements. They tasked the LLM to compare system goals expressed in Goal-oriented Language (GRL) with a set of security requirements and report back the exact nodes that address the requirements. Although the results were promising, the study focussed on software engineering requirements, not regulatory compliance contexts.

Rodriguez et al. [34] explore prompting strategies for explainable trace-link prediction for a dataset based on NASA's CM1 system. They observed promising results when they tasked an LLM with classifying and ranking requirement traceability. However, they report the occurrence of several false positives and note that they had to struggle to arrive at an optimal prompt configuration to get satisfactory results.

Prior work in requirements traceability shows that LLMs can recover trace links, but challenges related to accuracy and prompt sensitivity remain. This informs the traceability evaluation framework used in this study.

2.9 Terminology

This section defines the key terms used throughout this thesis.

Gap detection The process of identifying which specific regulatory clauses are not adequately addressed by an organisation's existing controls, mitigations, or policies. Gap detection involves both identifying the relevant clause and assessing the degree of coverage.

Coverage assessment The evaluation of whether an organisation's documented mitigation and control statements satisfy the requirements of a specific regu-

latory clause. Coverage is classified as *not covered*, *partially covered*, or *fully covered*.

Traceability The ability to link each compliance finding (gap, recommendation, or assessment) to a specific regulatory clause and supporting evidence. Traceability enables auditability by showing which clause justifies each finding.

Grounding The technique of providing an LLM with external evidence (retrieved documents) at inference time to ensure that its outputs are based on verifiable sources rather than parametric knowledge alone.

Parametric knowledge The information encoded in an LLM's weights during pretraining. This knowledge may be incomplete, outdated, or inaccurate for specialised domains.

Golden Standard Reference (GSR) An expert-validated dataset of 93 compliance risks with their correct regulatory clause mappings, expected coverage levels, and reference mitigations. The GSR serves as the ground truth for evaluating pipeline outputs.

Regulatory clause A specific numbered requirement within a regulatory framework (e.g., ISO/IEC 27002 control A.8.25 or ENISA clause 6.6.1) that specifies a control or measure an organisation must implement.

Mitigation A documented action, process, or measure that an organisation implements to address or reduce a compliance risk.

Remediation recommendation A generated suggestion for improving an organisation's risk description, mitigation, or control statement to better address a regulatory requirement.

3

Related Work

This chapter reviews prior work relevant to this thesis. Section 3.1 reviews studies on LLM-based compliance automation. Section 3.2 surveys recent RAG-based approaches for compliance and requirements engineering. Section 3.3 identifies the research gap addressed by this thesis.

3.1 LLM-Based Compliance Automation

Usman et al. [35] report an industrial case study on how compliance requirements are checked and analysed in large-scale software development at Ericsson. Their study shows that compliance analysis is manual and engineers need to interpret large volumes of documentation, as well as that mapping requirements to evidence is complex. Their findings highlight practical difficulties in compliance checking and analysis in a large organisation, motivating research that evaluates under realistic conditions. They emphasise that while regulatory compliance has a large body of proposed methods, there is comparatively limited empirical evidence describing the challenges faced in real industrial practice.

Hassani [18] explores the application of Large Language Models (LLMs) for automating the extraction of requirement-related legal content in the food safety domain and checking legal compliance, specifically GDPR. They apply prompt engineering and fine-tuning and report promising results such as relevant regulatory clauses and assess compliance conditions. Their results show that LLMs can extract regulatory/legal related text, identify relevant compliance requirements and support compliance checking. However, the paper was tested in a specific regulatory domain (food safety) and primarily relied on prompt-based extraction, leaving open questions regarding the reliability and traceability of LLM outputs in broader regulatory compliance settings. Also, they note that they would continue the research by applying RAG for GDPR retrieval.

In their pilot study, Salman et al. [19] assessed the utility of Sonnet 3.5, Llama 3.1, and ChatGPT 4o in the Audit Planning phase of the ISO 27001 framework for a hypothetical mobile operator HelloPak. The authors got promising results for activities such as Statement of Applicability generation, audit scope definition, and compliance checklist development. However, they noticed issues in handling

large data volumes and complex scenarios, such as correlating information across multiple documents and completeness issues with the LLMs skipping some clauses of the framework. They also came across false positives and noted that these issues limit the practical application of LLMs. They express the challenges in regulatory compliance and motivate the intention to evaluate more techniques, such as RAG.

A small-scale experiment by Chin et al. [32] explored security audit automation by experimenting on a specific task of auditing password policy compliance. Their promising results showed that LLMs could analyse policy text and identify whether security requirements are satisfied. However, the study focussed on a narrow compliance scenario and they did not consider the traceability of compliance claims. This is relevant to this thesis because the study examines both compliance gaps and ensures the results are traced back to the correct evidence.

In their 2023 study, Li et al. [13] compared the performance of ChatGPT and Gemini in providing cybersecurity advice in GRC settings. They queried the LLMs for Risk Assessment, Incident Response, Regulatory Compliance, and Threat Mitigation advice with questions ranging from open-ended to specific scenario-based questions inspired by real-world scenarios. They found that ChatGPT outperformed Gemini through a qualitative analysis evaluating the relevance, accuracy, completeness, and contextual appropriateness of its responses. However, they note that both models struggled with issues such as over-generalisation and false positives.

A survey by Esposito et al. [15], which included insights from researchers, practitioners, and policymakers in the IT governance domain, highlighted the interest in using AI systems to support governance and compliance tasks. The authors emphasised that AI systems must be transparent and explainable when used in governance and regulatory contexts—decisions must be defensible and auditable. These findings highlight the need for transparent and explainable implementation of LLM-based compliance automation.

3.2 RAG for Compliance and Requirements Engineering

A growing body of recent work has applied Retrieval-Augmented Generation (RAG) to compliance checking and requirements engineering tasks across several domains.

Das et al. [45] propose a multi-agent RAG framework for regulatory compliance checking of software requirements. Their system uses multiple agents with specialised retrieval roles to check whether software requirements satisfy regulatory constraints. While the framework demonstrates the potential of agentic RAG architectures for compliance tasks, it targets software requirements specifications rather than organisational GRC artifacts, and does not evaluate gap detection or recommendation generation.

Masoudifard et al. [46] combine Graph-RAG with advanced prompt engineering

techniques, such as Chain of Thought and Tree of Thought, to enhance requirement traceability and compliance checks in regulated environments such as finance and aerospace. They report significant improvements over baseline RAG and simple prompting strategies. However, the approach is costly and complex to implement, relies on having complete and accurate input data, and does not address regulatory gap analysis or the information security domain.

Sun et al. [47] present a compliance checking framework based on RAG, demonstrating that retrieval-augmented approaches can improve the accuracy of compliance verification. Their work focuses on general compliance checking (pass/fail determination) rather than gap detection with structured recommendations, and does not evaluate traceability to specific regulatory clauses.

In the medical device domain, Han et al. [48] introduce a RAG-based framework for standard applicability judgment with traceable justifications across multiple jurisdictions. Their system classifies standards as Mandatory, Recommended, or Not Applicable for a given device description. While the framework evaluates traceability, it addresses standard applicability determination rather than gap analysis, and operates in the medical device regulatory domain rather than information security.

Balu et al. [49] explore agent-based RAG for safety requirements derivation in autonomous driving, using a document pool of automotive standards. They show that the agent-based approach retrieves more relevant information compared to default RAG methods. However, their focus is on safety requirements derivation in the automotive domain, and they do not evaluate compliance gap detection or traceability to specific regulatory clauses.

Niu et al. [50] apply RAG to validate and recover traceability links between stakeholder and system requirements on industrial automotive data. Their approach (TVR) demonstrates practical effectiveness for traceability validation and recovery in complex automotive systems. While the study evaluates traceability with industrial data, it focuses on requirements traceability link recovery rather than regulatory gap detection, and operates in the automotive domain rather than information security GRC.

Arora et al. [51] use RAG for requirements extraction from domain documents in the space industry. Their work demonstrates that RAG can support the extraction of requirements from unstructured documentation. However, the task (requirements extraction) and domain (space industry) are different from the regulatory gap analysis in information security addressed in this thesis.

Khalid and Uygun [52] report an industrial-scale RAG evaluation on 669 automotive manufacturing requirements, achieving 98.2% extraction accuracy with complete traceability. Their study provides strong evidence that RAG can scale to industrial requirements engineering tasks. However, the focus is on requirements extraction and supplier qualification in automotive manufacturing, not on regulatory gap detection or recommendation generation in information security.

Allani et al. [53] propose ELISAR, a multi-agent RAG framework for cybersecurity operations including a GRC component, though its compliance module focuses on general checking rather than gap analysis.

3.3 Research Gap

The studies reviewed above demonstrate the potential of LLMs and RAG in compliance and requirements engineering contexts, but also highlight recurring issues such as hallucination, false positives, and limited evaluation in industrial settings. Recent RAG-based approaches have been applied to compliance checking, traceability recovery, requirements extraction, and standard applicability determination across domains including automotive, medical devices, aerospace, and software engineering.

However, none of these studies address *regulatory gap analysis* in the information security domain—specifically, identifying missing or partially implemented controls against regulatory clauses and generating actionable remediation recommendations. Furthermore, no prior work jointly evaluates the precision, recall, and clause-level traceability of LLM-generated gap analysis outputs across the full scope of information security controls under frameworks such as ENISA (NIS2) and ISO/IEC 27001. This gap motivates the empirical evaluation conducted in this thesis.

There are alternatives to RAG were considered for addressing this gap:

- **Fine-tuning** would require labelled compliance datasets that are scarce and proprietary in the information security domain, and would need to be repeated for each new regulatory framework or model version. Ovadia et al. [61] show that RAG consistently outperforms unsupervised fine-tuning for knowledge injection, both for existing and new knowledge, and that LLMs struggle to learn new factual information through fine-tuning alone.

- **Long-context models** could theoretically ingest entire regulatory documents, but this approach does not scale when multiple frameworks must be queried dynamically, and it lacks the explicit retrieval evidence chain needed for auditable compliance findings. Recent studies show that LLMs struggle to utilise information in long contexts effectively [31] and that context length alone hurts performance even with perfect retrieval [22].

- **Agentic architectures** (multi-agent RAG) are emerging [45, 62] but add implementation complexity—requiring routing, scheduling, and fusion mechanisms—without addressing the fundamental question of whether inference-time grounding improves gap analysis quality.

RAG was selected because it enables traceable, auditable outputs without model modification—a critical requirement in regulatory contexts where each finding must be defensible and linked to specific regulatory text [15]. Additionally, RAG remains the most widely adopted and cost-effective grounding technique for knowledge-intensive tasks in production systems [25].

4

Methods

This chapter describes the experimental methodology following the framework proposed by Wohlin et al. [27]. Section 4.1 presents the research design and goals. Section 4.2 defines the experimental variables and hypotheses. Section 4.3 describes the empirical context and justifies the experimental setting. Section 4.4 details the experimental procedure. Section 4.5 describes the pipeline implementation. Section 4.6 presents the prompt design. Section 4.7 explains the evaluation framework, including expert evaluation and statistical analysis.

4.1 Research Design

This study is designed as a controlled computational experiment in which baseline and retrieval-augmented LLM configurations are applied to the same dataset under identical conditions. The goal is to evaluate how inference-time grounding affects the performance of LLM-assisted regulatory gap analysis, with a primary focus on comparing a non-grounded baseline against a retrieval-augmented configuration.

The study follows Wohlin et al.'s [27] experimentation framework, which structures empirical software engineering experiments into five phases: scoping, planning, operation, analysis, and interpretation. This framework was chosen because it provides a systematic approach to defining variables, controlling confounds, and ensuring reproducibility—all essential for a comparative evaluation of LLM configurations.

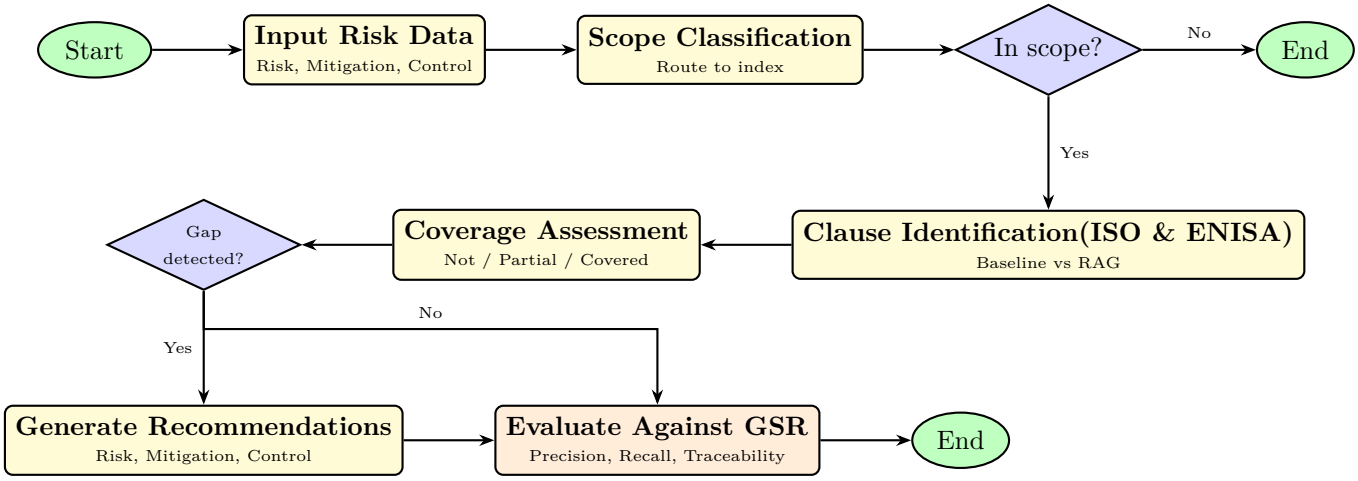

Figure 4.1: Pipeline of the LLM-assisted regulatory gap analysis system. The system processes risk inputs, classifies scope, identifies regulatory clauses (baseline or RAG), assesses coverage, generates recommendations for identified gaps, and evaluates outputs against the GSR using precision, recall, and traceability.

Figure 4.1 illustrates the overall structure of the computational experiment. The preparation phase includes dataset construction, GSR development, prompt design, LLM configuration, and framework development. The computational experimentation phase generates compliance assessment outputs by comparing non-grounded baseline against retrieval-augmented generation. The final phase analyses these outputs using precision, recall, and traceability, and links the results to RQ1, RQ2, and RQ3.

4.2 Experimental Variables and Hypotheses

Table 4.1 defines the experimental variables.

Table 4.1: Experimental variables

Category	Description
Primary independent variable	Grounding configuration: baseline (no retrieval) vs. retrieval-augmented generation (RAG with hybrid search)
Secondary independent variable	LLM model choice: GPT-5.1 vs. Claude Opus 4.6
Dependent variables	Clause prediction accuracy (exact match, weighted, top-5 recall), coverage assessment accuracy (strict, relaxed, Spearman's ρ), mitigation quality (semantic recall, LLM-as-Judge scores), and traceability (clause-level linking)
Controlled variables	Dataset (93 risks), prompt structure, top-k retrieval depth ($k = 5$), evaluation protocol, regulatory corpus (ISO/IEC 27002 + ENISA NIS2), embedding model (`text-embedding-3-large`), temperature (0)
Unit of analysis	Individual compliance risk together with its mapped clause(s), coverage decision, and mitigation output

The *primary independent variable* is the grounding configuration: whether the LLM receives retrieved regulatory context (RAG) or relies solely on parametric knowledge (baseline). The *secondary independent variable* is the LLM model, included to assess whether performance differences generalise across architectures.

The *dependent variables* measure three aspects of gap analysis quality: (1) accuracy of clause identification and coverage assessment, (2) quality of generated remediation recommendations, and (3) traceability of outputs to specific regulatory clauses.

4.2.1 Hypotheses

RQ1 – Gap detection:

H_0: There is no difference in gap detection performance (clause prediction accuracy, coverage assessment accuracy) between the baseline and retrieval-augmented configurations.

H_1: Retrieval-augmented generation improves gap detection performance compared to the non-grounded baseline.

RQ2 – LLM model effect:

H_0: The choice of LLM model does not affect gap detection performance under identical experimental conditions.

H_1: The choice of LLM model affects gap detection performance.

RQ3 – Mitigation quality:

H_0: There is no difference in mitigation quality (NLP metrics, LLM-as-Judge evaluation) between the non-grounded baseline and the retrieval-augmented configuration.

H_1: There is a difference.

Each hypothesis is tested using McNemar's exact binomial test (for paired categorical outcomes) and the Wilcoxon signed-rank test (for paired continuous metrics), with significance level $\alpha = 0.05$.

4.3 Empirical Context and Setting

The empirical setting for this study is Stratsys, a Swedish company that provides a cloud-based platform for governance, risk, and compliance (GRC) management. Stratsys was chosen as the experimental context for three reasons: (1) it provides access to realistic organisational risk data representative of the information security domain; (2) the company's GRC experts contributed domain knowledge for constructing and validating the Golden Standard Reference; and (3) the platform context ensures the experiment evaluates gap analysis under conditions representative of industrial GRC workflows, where organisational artifacts such as risk descriptions, mitigations, and controls must be mapped against regulatory frameworks.

Within the company, regulatory compliance is performed by comparing organisational policies, controls, and evidence against regulatory requirements such as ISO/IEC 27001. In practice, this process involves manual analysis of large volumes of documents. Compliance officers must verify whether organisational artifacts satisfy regulatory clauses and provide sufficient evidence. The experiment evaluates whether LLM-based automation can support this process with improved accuracy and traceability.

4.4 Experimental Procedure

The experiment follows a structured process consisting of three phases: preparation, execution, and evaluation.

4.4.1 Dataset Construction

The experiment dataset consists of 93 compliance risks representative of information security governance scenarios. Each risk is a structured record containing:

- A risk description (mandatory) — a statement identifying a potential security weakness or non-compliance.

- A mitigation statement (optional) — documented actions the organisation has taken to address the risk.
- A control statement (optional) — the formal control implemented by the organisation.

The 93 risks are distributed across three coverage groups to enable systematic evaluation of coverage assessment accuracy:

- **Group 0** (31 risks): No mitigation or control provided. Simulates risks where the organisation has not yet addressed the regulatory requirement.
- **Group 1** (31 risks): Partial mitigation provided. Simulates risks where some measures exist but gaps remain.
- **Group 2** (31 risks): Full mitigation and control provided. Simulates risks where the organisation has fully addressed the requirement.

This balanced design enables evaluation of the pipeline's ability to discriminate between different coverage levels. The risks span all four ISO 27002 control themes (organisational, people, physical, technological controls, chapters 5–8), providing broad coverage of the standard's control structure.

4.4.2 Golden Standard Reference (GSR)

The Golden Standard Reference (GSR) was created as the ground truth for evaluating pipeline outputs. For each of the 93 risks, the GSR specifies:

- The correct ISO/IEC 27002 clause (e.g., A.8.25).
- The correct ENISA NIS2 clause (e.g., 6.6.1).
- The expected coverage level (not covered, partially covered, or covered).
- Reference mitigation and control statements that represent adequate compliance.

The GSR was constructed by the same information security expert (Expert A) who identified the risks. For each risk, the expert defined the correct regulatory clause mappings, expected coverage levels, and reference mitigations based on professional domain knowledge of both ISO/IEC 27002 and ENISA NIS2 requirements. This ensures that the GSR reflects expert-validated ground truth directly grounded in practitioner knowledge.

4.4.3 Experimental Configurations

Four experimental configurations were evaluated in a 2×2 factorial design:

Table 4.2: Experimental configurations

	Baseline (no retrieval)	**RAG (hybrid retrieval)**
GPT-5.1	Config 1	Config 2
Claude Opus 4.6	Config 3	Config 4

All four configurations use the same dataset, prompt structure, and evaluation protocol. The only differences are the independent variables (grounding configuration and LLM model). Temperature was set to 0 for all runs to minimise non-determinism.

Two LLM models were selected to assess generalisability:

- **GPT-5.1** (OpenAI): Selected because it is available within the company's Azure environment and represents a state-of-the-art commercial model.
- **Claude Opus 4.6** (Anthropic): Selected to provide a cross-architecture comparison and because it represents an alternative leading model with different pre-training characteristics.

4.4.4 Data Collection Procedure

Each configuration was executed on all 93 risks. For each risk, the pipeline produced structured outputs including: scope classification, clause selection with evidence, coverage assessment with reasoning, and (when gaps were detected) improved risk, mitigation, and control recommendations. All outputs were stored as structured JSON records for subsequent analysis.

4.5 Pipeline Implementation

The pipeline processes each input risk through four sequential stages. Each stage is implemented as a separate LLM call with a task-specific prompt, ensuring that intermediate outputs are independently traceable and evaluable.

Stage 1: Scope Classification. The input risk, mitigation, and control text are classified into one of four categories: general information security (ENISA-scope), secure software development lifecycle (ISO SDLC-scope), incident management, or multiple categories. This routing step determines which regulatory index is queried in the subsequent retrieval stage. Out-of-scope items are filtered and excluded from further processing.

Stage 2: Clause Identification. In the baseline configuration, the LLM identifies the most relevant regulatory clause using only its parametric knowledge. In the RAG configuration, the system first retrieves the top-5 most relevant clauses from the regulatory index using hybrid search (BM25 + vector similarity via Azure AI Search with the `text-embedding-3-large` embedding model). The LLM then selects the

single best-matching clause from among the retrieved candidates, providing a reason and exact supporting evidence from the clause text. This constraint—selecting only from retrieved context—prevents hallucination of non-existent clause identifiers.

Stage 3: Coverage Detection. The LLM compares the organisation's mitigation and control statements against the selected regulatory clause to determine the coverage level. Three levels are defined: *not covered* (no meaningful coverage), *partially covered* (some aspects addressed but clear gaps remain), and *covered* (mitigation and control fully satisfy clause requirements). The coverage assessment includes a structured reason explaining what is or is not covered.

Stage 4: Recommendation Generation. When the coverage assessment identifies a gap (not covered or partially covered), the LLM generates improved risk, mitigation, and control statements that would fully satisfy the identified regulatory clause. The generated recommendations are structured as JSON fields to enable systematic evaluation against the GSR.

4.6 Prompt Design

The prompts were developed through iterative refinement over multiple development cycles. Each prompt uses three techniques in combination: role prompting (assigning a specific persona), structured output constraints (requiring JSON format), and context restriction (constraining the model to use only provided information).

The full prompt specifications, including all iterations and the final versions used in the experiment, are provided in Appendix A.

4.6.1 Design Principles

The following principles guided prompt design:

1. **Separation of concerns:** Each pipeline stage has a dedicated prompt with a single, well-defined objective. This prevents task interference and enables independent evaluation of each stage.

2. **Context restriction:** In the RAG configuration, the clause selection prompt explicitly constrains the model to select only from retrieved candidates ("Use ONLY the given context"), preventing hallucination of clause identifiers.

3. **Structured output:** All prompts require JSON-formatted responses with predefined fields, ensuring outputs are machine-parsable and consistently structured across experimental runs.

4. **Decision rules:** The coverage detection prompt provides explicit rules for each coverage level to reduce subjective interpretation and improve reproducibility.

4.7 Evaluation Framework

The evaluation framework measures pipeline performance across three dimensions: clause prediction accuracy, coverage assessment accuracy, and mitigation quality.

4.7.1 Clause Prediction Metrics

Clause prediction is evaluated by comparing the pipeline's selected clause against the GSR's expected clause. Three levels of match are defined:

- **Exact match:** The predicted clause identifier matches the GSR clause exactly (e.g., both specify A.8.25).
- **Sibling match:** The predicted clause is within the same parent control family (e.g., A.8.24 predicted when A.8.25 is expected). This captures near-misses where the correct control area was identified.
- **Miss:** The predicted clause belongs to a different control family or is incorrect.

From these, three metrics are computed: exact match accuracy (proportion of exact matches), weighted accuracy (exact = 1.0, sibling = 0.5, miss = 0.0), and top-5 recall (whether the correct clause appears in the retrieved candidate set, applicable only to RAG configurations).

4.7.2 Coverage Assessment Metrics

Coverage assessment is evaluated under two criteria:

- **Strict accuracy:** The predicted coverage level exactly matches the GSR coverage level.
- **Relaxed accuracy:** Adjacent predictions are accepted (e.g., predicting "partially covered" when the ground truth is "covered" is counted as correct). This accounts for the inherent subjectivity in coverage boundaries.

Additionally, Spearman's rank correlation (ρ) is computed between the predicted and expected coverage levels (encoded as ordinal: not covered = 0, partially covered = 1, covered = 2) to assess whether the pipeline correctly ranks coverage severity.

4.7.3 Mitigation Quality Metrics

Mitigation quality is measured using both automated NLP metrics and LLM-as-Judge evaluation:

Automated metrics:

- **Semantic recall:** The proportion of GSR reference mitigation content that is semantically captured in the generated output, computed using sentence-level

embedding similarity with a threshold of 0.7.

- **BERTScore F1:** Token-level semantic similarity between generated and reference text using contextual embeddings.
- **ROUGE-1/ROUGE-L F1:** Surface-level lexical overlap metrics measuring unigram and longest common subsequence overlap.
- **Jaccard similarity:** Set-based word overlap between generated and reference mitigations.

LLM-as-Judge evaluation: Claude Opus 4.6 is used as an automated judge to evaluate mitigation quality on four dimensions (relevance, completeness, actionability, specificity) using a Likert scale (1–5), plus an overall quality score (1–10). The judge also performs pairwise comparison between baseline and RAG outputs. The model selection was restricted by the options approved for use by Stratsys. To avoid self-evaluation bias, the LLM-as-Judge uses a different model (Claude Opus 4.6) from the primary pipeline model being evaluated (GPT-5.1). This cross-model design ensures that the judge has no prior exposure to or preference for its own outputs. The three-method evaluation design (NLP metrics, LLM-as-Judge, and expert evaluation) provides triangulation—if any single method is biased, disagreement between methods exposes this.

4.7.4 Expert Evaluation

Expert evaluation was conducted to validate the automated evaluation methodology and provide human-assessed quality judgments. The study initially planned for two independent domain experts to evaluate pipeline outputs, enabling inter-rater reliability analysis. However, one evaluator was unable to contribute the evaluation within the project timeline. The study therefore proceeded with a single expert evaluator.

The evaluating expert (Expert B)—an information security professional with experience in ISO 27001 implementation—independently evaluated a stratified random sample of 30 risks per configuration (120 records total across 73 unique risks). A separate GRC expert at Stratsys (Expert A) constructed and validated the Golden Standard Reference, defining the correct clause-level traceability mappings for all 93 risks. Expert A did not perform the independent output evaluation, avoiding circular validation.

Although only Expert B performed the direct output evaluation, the study achieves indirect inter-rater reliability for traceability through the separation of roles. Expert A's judgment is encoded in the GSR (what constitutes a correct clause mapping), while Expert B independently assessed whether pipeline outputs match the correct clauses. When Expert B agrees with the automated evaluation—which uses Expert A's GSR as ground truth—this constitutes implicit agreement between the two experts on clause-level correctness. The resulting agreement ($\kappa = 1.00$ for ISO,

$\kappa = 0.97$ for ENISA) therefore reflects convergent validity between two independent domain experts, strengthening the traceability validation despite the single-evaluator design. In cases where the two experts disagree, disagreements are resolved through both experts independently reviewing the contested case and then discussing it in a separate meeting until consensus is reached. This consensus-based approach ensures that the final ground truth reflects shared expert judgment rather than a single rater's perspective.

The sample size of 30 per configuration was chosen based on two considerations: (1) it provides sufficient statistical power for the Mann-Whitney U test ($\beta \geq 0.80$ for detecting medium effect sizes at $\alpha = 0.05$); and (2) it represents approximately one-third of the full dataset, balancing evaluation thoroughness against expert time constraints. Stratified sampling across coverage groups (10 per group per configuration) ensures representativeness.

The expert evaluated each record on:

1. **Classification correctness** (Yes/No): Whether the predicted ISO and ENISA clauses match the golden standard.

2. **Quality ratings** (Likert 1–5): Relevance and actionability of generated mitigations, controls, and risk text, evaluated separately for ISO and ENISA outputs.

3. **Free-text observations**: Qualitative comments on output quality.

The purpose of expert evaluation is twofold: first, to validate that the automated exact-match evaluation accurately reflects human judgment (methodology validation); and second, to assess whether the generated recommendations are practically useful for compliance professionals (quality assessment). This addresses RQ3 by providing human-validated evidence of mitigation quality beyond what automated metrics can capture.

4.7.5 Statistical Tests

The following statistical tests are used to assess significance:

- **McNemar's exact binomial test:** Used for paired binary outcomes (e.g., correct vs. incorrect clause prediction). This test is appropriate because each risk is evaluated under both configurations (baseline and RAG), producing paired dichotomous observations. McNemar's test evaluates whether the number of discordant pairs (cases where one configuration is correct and the other is not) deviates significantly from chance.

- **Wilcoxon signed-rank test:** Used for paired continuous metrics (e.g., mitigation quality scores). This non-parametric test is appropriate because the metric distributions are non-normal and the observations are paired (same risk evaluated under both configurations).

- **Mann-Whitney U test:** Used for comparing independent samples in the expert evaluation (where sampling was performed independently per configuration).

- **Spearman's rank correlation:** Used to assess ordinal agreement between predicted and expected coverage levels.

All tests use $\alpha = 0.05$. Where multiple comparisons are performed, Bonferroni correction is applied and noted.

4.8 Threats to Validity

4.8.1 Internal Validity

- **Confounding variables:** The controlled experimental design (same dataset, prompts, and evaluation protocol across configurations) minimises confounds. Temperature is set to 0 to reduce non-determinism, though minor stochastic variation may still occur.

- **Instrumentation:** The GSR was constructed by an information security expert at Stratsys who defined the correct clause mappings for all 93 risks. The dataset itself consists of real-world risks identified by the same expert. Any errors in the GSR directly affect reported accuracy. To mitigate this, a second expert (Expert B) independently validated a subset of the GSR entries, achieving near-perfect agreement ($\kappa \geq 0.97$).

4.8.2 External Validity

- **Generalisability:** Results are specific to the information security domain (ISO/IEC 27002 and ENISA NIS2). Performance may differ for other regulatory frameworks or domains.

- **Model evolution:** Results are tied to specific LLM versions (GPT-5.1 and Claude Opus 4.6). Newer model versions may exhibit different performance characteristics.

4.8.3 Construct Validity

- **Metric validity:** Exact match accuracy may underestimate performance when the predicted clause is semantically correct but uses a different clause identifier. The weighted accuracy and sibling-match metrics partially address this.

- **Coverage subjectivity:** The boundary between "partially covered" and "covered" involves subjective judgment. The relaxed accuracy metric and Spearman's correlation mitigate this.

4.8.4 Reliability

- **Reproducibility:** All prompts, configurations, and evaluation scripts are documented. However, use of licensed ISO/IEC 27001 text and potentially confidential organisational artifacts may restrict full public release of evaluation data.
- **LLM non-determinism:** Despite temperature = 0, LLMs may produce slightly different outputs across runs. Single-run execution is used; repeated runs are left for future work.

4.9 Supplementary Material

The evaluation scripts, data processing pipelines, and prompt iterations used in this study are available in an external repository.[1] The supplementary material includes:

- Python scripts for clause prediction evaluation, coverage assessment, and mitigation quality metrics.
- LLM-as-Judge evaluation scripts.
- Prompt versions and iteration history.
- Raw evaluation output data in JSON format.

[1] Repository URL to be added upon publication.

5

Results

This chapter presents the experimental results across all four configurations. Section 5.1 reports descriptive statistics of the dataset and pipeline outputs. Section 5.2 presents clause identification and coverage assessment results (RQ1). Section 5.3 reports the effect of LLM model choice (RQ2). Section 5.4 reports mitigation recommendation quality (RQ3). Section 5.5 presents expert validation. Section 5.6 summarises statistical significance. Section 5.7 provides the overall findings.

5.1 Descriptive Statistics

Table 5.1 summarises the dataset composition. All 93 risks have both ISO/IEC 27002 and ENISA NIS2 ground-truth mappings in the GSR. The risks are distributed into three coverage groups of equal size (31 each) to enable balanced statistical comparison. This balanced design was chosen because equal group sizes maximise the statistical power of non-parametric tests and ensure that each coverage level contributes equally to accuracy metrics.

The risks span four ISO 27002 control themes. ISO/IEC 27002 organises its controls starting from Chapter 5 (Chapters 1–4 address scope, normative references, terms, and structure). The four control themes in Chapters 5–8 are: organisational controls (Ch. 5), people controls (Ch. 6), physical controls (Ch. 7), and technological controls (Ch. 8).

Table 5.1: Dataset composition

Characteristic	**Value**
Total risks	93
Group 0 (no mitigation provided)	31
Group 1 (partial mitigation provided)	31
Group 2 (full mitigation provided)	31
ISO Ch. 5 Organisational	37 (39.8%)
ISO Ch. 6 People	8 (8.6%)
ISO Ch. 7 Physical	14 (15.1%)
ISO Ch. 8 Technological	34 (36.6%)
Risks with ENISA mapping	93 (100%)

Table 5.2 summarises key characteristics of the generated outputs. The *confidence score* is a self-reported value (scale 1–10) that the LLM assigns to its own clause selection, indicating how certain it is that the chosen clause is the correct match. This score is extracted from the structured JSON output of the clause selection step.

Table 5.2: Descriptive statistics of pipeline outputs across all four configurations

Metric	GPT-5.1 B	GPT-5.1 RAG	Claude B	Claude RAG
Confidence scores (1–10)				
Mean (SD)	8.73 (0.61)	9.23 (0.64)	9.41 (0.58)	9.14 (0.92)
Median	9.0	9.0	9.0	9.0
Range	7–10	7–10	8–10	5–10
Coverage predictions				
Not covered	51	32	46	38
Partially covered	36	61	39	54
Covered	6	0	8	1
Mitigation output (word count)				
n (non-empty)	87	88	80	91
Mean (SD)	342 (78)	395 (77)	232 (48)	272 (54)
Median	329	401	224	266
Range	214–588	238–551	144–436	160–452
Mitigation items per risk				
Mean (Median)	11.6 (12)	13.3 (14)	10.0 (10)	11.0 (11)
Range	8–18	9–20	7–14	1–20

Several patterns emerge from the descriptive statistics. First, all configurations exhibit high self-reported confidence (median = 9 across all four), with limited variance, suggesting potential overconfidence. Second, both pipelines show a strong conservative bias in coverage assessment—predicting `covered` in fewer than 9% of cases despite one-third of inputs having full mitigations. Third, RAG configurations produce longer mitigations (395 vs 342 words for GPT-5.1; 272 vs 232 for Claude), consistent with grounding in verbose regulatory text. Fourth, GPT-5.1 generates substantially more verbose output than Claude across both conditions.

5.2 Clause Identification and Coverage Assessment (RQ1)

This section evaluates whether retrieval-augmented grounding improves the accuracy of regulatory clause identification and coverage assessment compared to the non-grounded baseline.

5.2.1 Consolidated Clause Prediction Results

Table 5.3 presents all clause prediction results across both models, both configurations, and both regulatory frameworks. The table is organised by document (ISO 27002 and ENISA NIS2), with columns grouped by grounding configuration (RAG vs Baseline) and LLM model within each configuration.

Table 5.3: Consolidated clause prediction accuracy ($n = 93$). Bold values indicate the best result per metric. Grey cells highlight statistically significant differences between LLMs.

	RAG		**Baseline**		**LLM diff.**
Metric	GPT-5.1	Claude	GPT-5.1	Claude	p-value
Document 1: ISO/IEC 27002					
Exact match	79.6%	**84.9%**	43.0%	**86.0%**	$<$ **0.001***** (B)
Weighted accuracy	0.860	**0.887**	0.629	**0.887**	—
Top-5 recall	95.7%	95.7%	—	—	= (same index)
Sibling matches	12.9%	7.5%	39.8%	5.4%	—
Complete misses	7.5%	7.5%	17.2%	8.6%	—
B vs RAG (p)	GPT: 0.000001***		Claude: 1.0 (ns)		
LLM diff (p)	0.125 (ns)		$<$ **0.001*****		
Document 2: ENISA NIS2					
Exact match	**33.3%**	35.5%	0.0%	0.0%	ns
Top-5 recall	63.4%	63.4%	—	—	= (same index)
B vs RAG (p)	$<$ 0.001***		$<$ 0.001***		

The consolidated view reveals several key findings. For ISO clause prediction, RAG substantially improves GPT-5.1 (from 43.0% to 79.6%, $p = 0.000001$) but does not significantly improve Claude (86.0% to 84.9%, $p = 1.0$). Claude achieves high accuracy from parametric knowledge alone, suggesting that its pre-training data includes extensive ISO 27002 content. The majority of GPT-5.1 baseline errors are sibling-level near-misses (39.8%), indicating that the model identifies the correct control family but selects the wrong specific control within it.

For ENISA clause prediction, neither model can identify clauses from parametric knowledge (0% for both baselines), suggesting that the ENISA NIS2 Technical Implementation Guidance (published June 2025) is not represented in pre-training data. RAG enables moderate ENISA accuracy (33–35%), with the retrieval step surfacing the correct clause in the top-5 candidates in 63.4% of cases. The gap between top-5 recall and final selection accuracy suggests room for improvement in the generation step.

Both RAG configurations achieve identical top-5 recall (95.7% ISO, 63.4% ENISA), confirming that retrieval performance is independent of the generation model—both share the same index and embedding model.

5.2.2 Coverage Assessment Results

Table 5.4 presents coverage assessment accuracy following the same structure, grouped by configuration and model.

Table 5.4: Coverage assessment accuracy ($n = 93$). Bold values indicate the best per-metric result. Statistically significant differences ($p < 0.05$) are marked with asterisks.

	RAG		**Baseline**	
Metric	GPT-5.1	Claude	GPT-5.1	Claude
Strict accuracy	65.6%	62.4%	59.1%	62.4%
Relaxed accuracy	**98.9%**	92.5%	78.5%	83.9%
Spearman's ρ	**0.859**	0.778	0.531	0.694
Relaxed accuracy by input group (GPT-5.1 only)				
Group 0 (no mitigation)	100.0%	—	100.0%	—
Group 1 (partial mitigation)	96.8%	—	71.0%	—
Group 2 (full mitigation)	100.0%	—	64.5%	—
Statistical tests (McNemar's, Baseline vs RAG)				
GPT-5.1	$\mathbf{p = 0.000004}^{***}$			
Claude	$p = 0.057$ (ns)			

Under the relaxed evaluation criterion, RAG achieves near-perfect coverage assessment for GPT-5.1 (98.9%, $p = 0.000004$). Spearman's rank correlation ($\rho = 0.859$) confirms strong ordinal agreement—when more mitigations are provided in the input, RAG reliably rates coverage higher. For Claude, RAG improves coverage assessment from 83.9% to 92.5%, but this difference does not reach statistical significance ($p = 0.057$). This is because Claude's baseline performance is already relatively strong—its parametric knowledge of ISO 27002 provides a higher starting accuracy, leaving less room for RAG to demonstrate statistically significant improvement. The smaller effect size (8.6 percentage points vs. 20.4 for GPT-5.1) combined with the sample size ($n = 93$) results in insufficient statistical power to detect this smaller difference at $\alpha = 0.05$. Both pipelines exhibit conservative bias (rarely predicting `covered`), which is arguably a safer failure mode in compliance contexts.

5.2.3 Confidence Calibration

RAG produces better-calibrated confidence scores for GPT-5.1: higher correlation between confidence and actual accuracy ($r = 0.511$ vs 0.349), more high-confidence correct predictions, and fewer high-confidence errors (18 vs 49 cases where confidence ≥ 7 but prediction is wrong). The baseline is overconfident—it assigns high confidence scores even when predictions are incorrect.

5.2.4 Summary: RQ1

RAG significantly improves clause identification and coverage assessment for GPT-5.1. For Claude, which already possesses strong parametric knowledge of ISO 27002, RAG provides only a small, non-significant improvement. For both models, RAG is essential for ENISA clause identification, which is impossible from parametric knowledge alone.

5.3 Effect of LLM Model Choice on Gap Detection Performance (RQ2)

This section presents the results comparing GPT-5.1 and Claude Opus 4.6 on clause prediction accuracy and coverage assessment under both configurations.

Without retrieval, Claude Opus 4.6 substantially outperforms GPT-5.1 on ISO clause prediction (86.0% vs 43.0%, $p < 0.000001$). This 43 percentage point gap reflects differences in parametric knowledge of the ISO 27002 framework between the two models. Neither model can identify ENISA clauses from parametric knowledge.

Under RAG, the ISO clause prediction gap narrows from 43 pp to 5.3 pp and is no longer statistically significant ($p = 0.125$). Both models achieve identical top-5 recall, confirming that retrieval performance is model-independent. For coverage assessment, GPT-5.1 RAG achieves slightly higher accuracy than Claude RAG (98.9% vs 92.5%, $p = 0.031$), though this does not survive Bonferroni correction.

Table 5.5: Performance gap between models: baseline vs RAG

Metric	Gap (Baseline)	Gap (RAG)
ISO exact match	43.0 pp	5.3 pp
Weighted accuracy	0.258	0.027
Coverage (relaxed)	5.4 pp	6.4 pp†
†GPT-5.1 RAG is higher for coverage		

RAG reduces the inter-model performance gap from 43 pp to 5.3 pp for clause prediction. This suggests that retrieval-augmented grounding compensates for weaker

parametric knowledge, reducing dependency on specific model capabilities. Organisations could select LLMs based on cost or availability rather than domain expertise, provided RAG infrastructure is in place.

5.4 Quality of Generated Remediation Recommendations (RQ3)

This section evaluates whether retrieval-augmented grounding produces higher-quality mitigation recommendations. The primary comparison is presented for ISO/IEC 27002 outputs (GPT-5.1 model), because pairwise mitigation comparison requires both configurations to identify a valid clause—since the ENISA baseline achieves 0% clause accuracy (Section 5.2), its mitigations are generated against incorrect clauses, making direct quality comparison uninformative. ENISA mitigation quality under RAG is discussed separately where noted. For fair pairwise comparison, only risks where both configurations produce non-empty mitigations are included ($n = 83$ for GPT-5.1). Ten risks were excluded because at least one configuration produced an empty mitigation field, predominantly in Group 2 where the pipeline correctly identified full coverage and did not generate redundant recommendations.

5.4.1 Comparison of Evaluation Methods

Three independent methods evaluated mitigation quality, yielding partially divergent results. Table 5.6 presents the three evaluation methods side by side.

Table 5.6: Mitigation quality: comparison of three evaluation methods (GPT-5.1, $n = 83$)

Method	**Baseline**	**RAG**	**Finding**
NLP metrics (automated, lexical overlap)			
Semantic recall	0.539	0.542	No difference (ns)
BERTScore F1	0.171	0.158	Baseline higher ($p = 0.008$)
ROUGE-L F1	0.054	0.046	Baseline higher ($p < 0.001$)
LLM-as-Judge (Claude evaluating GPT-5.1 outputs)			
Relevance (1–5)	4.48	**4.78**	RAG higher
Completeness (1–5)	4.28	**4.61**	RAG higher
Actionability (1–5)	4.70	**4.98**	RAG higher
Overall (1–10)	7.71	**8.49**	RAG higher
Pairwise wins	16.9%	**80.7%**	RAG wins
Expert evaluation (ISO mitigations, Likert 1–5)			
Mean (Median)	4.45 (5.0)	4.55 (5.0)	No difference ($p = 0.396$)

The three methods disagree on ISO mitigation quality, and this disagreement is itself an important finding. The divergence is explainable by the distinct properties of each method:

NLP metrics measure lexical overlap with the golden standard reference. RAG produces longer, more framework-aligned text that uses different vocabulary from the golden standard's concise phrasing, depressing lexical overlap scores without reducing semantic coverage. These metrics are therefore not well-suited for evaluating generative compliance outputs where correct answers can be expressed in many different ways.

The LLM-as-Judge evaluates relevance, completeness, actionability, and specificity—dimensions where RAG's grounding in retrieved regulatory text provides measurable advantage through richer contextual detail. The judge (Claude Opus 4.6) evaluates GPT-5.1 outputs to avoid self-evaluation bias.

The expert evaluation shows a ceiling effect: both pipelines achieve median quality of 5/5 for ISO mitigations, leaving insufficient variance to detect differences. The

expert's holistic "good enough" threshold is reached by both pipelines for ISO-grounded content.

For ENISA outputs, all three methods converge: RAG produces significantly higher-quality content ($p < 0.001$ on expert ratings), because baseline cannot produce meaningful ENISA-grounded output at all.

5.4.2 Misleading Outputs

RAG produces fewer misleading mitigations—records where the pipeline generated substantial output (≥5 items) but achieved low semantic recall (<0.4): 2 records (2.4%) for RAG vs 7 records (8.4%) for baseline. This indicates that grounding in retrieved regulatory text reduces the risk of generating plausible-sounding but off-target recommendations.

5.5 Expert Validation

A domain expert independently evaluated a stratified random sample of 30 risks per configuration (120 records total). The evaluation validates the automated methodology and provides human-assessed quality judgments.

5.5.1 Agreement Between Expert and Automated Evaluation

Table 5.7: Expert vs automated evaluation agreement ($n = 120$)

Framework	Agreement	Cohen's κ	Concordant correct	Concordant wrong
ISO 27002	100.0% (120/120)	1.000	88	32
ENISA NIS2	99.2% (119/120)	0.967	18	101

The expert and automated evaluation achieve perfect agreement on ISO clause correctness ($\kappa = 1.00$) and near-perfect agreement on ENISA ($\kappa = 0.97$). Cohen's κ is computed here between Expert B's binary verdict (correct/incorrect) and the automated evaluation's binary verdict on the same 120 records, treating the two assessment methods as independent raters of the same items.

Importantly, because the automated evaluation uses Expert A's GSR as ground truth, this agreement also represents indirect inter-rater reliability between the two domain experts. Expert A defined what constitutes a correct clause mapping (via the GSR), and Expert B independently confirmed those same correctness judgments when evaluating pipeline outputs. The near-perfect convergence ($\kappa \geq 0.97$) between their judgments—despite being made independently and at different stages of the study—provides strong evidence that the traceability mappings are robust and not artefacts of a single evaluator's interpretation.

5.5.2 Expert Quality Ratings by Framework

For ISO-grounded outputs, the expert rates both pipelines highly (median mitigation quality = 5/5) with no statistically significant difference between baseline and RAG (Mann-Whitney U, $p = 0.396$). For ENISA-grounded outputs, RAG produces significantly higher-quality content on all dimensions ($p < 0.001$), while baseline ENISA ratings are at floor (median = 1) because baseline configurations cannot identify correct ENISA clauses.

5.5.3 Qualitative Observations

The expert provided 11 free-text comments. Three themes emerged: (1) generated risk text was correct but excessively verbose (7/11 comments); (2) one case of semantic shift where the improved risk statement altered the original risk meaning; (3) two records with empty outputs for Group 2 risks where coverage was correctly assessed as sufficient.

5.6 Statistical Significance Summary

Table 5.8 summarises all statistical tests performed. Bonferroni-corrected significance threshold: $\alpha_{\text{adj}} = 0.05/16 \approx 0.003$.

Table 5.8: Statistical significance summary (all comparisons)

Comparison	**Metric**	**Test**	**p-value**
RQ1: RAG vs Baseline			
GPT-5.1: B vs RAG	ISO exact match	McNemar	0.000001^{***}
GPT-5.1: B vs RAG	ENISA exact match	McNemar	$< 0.000001^{***}$
GPT-5.1: B vs RAG	Coverage (relaxed)	McNemar	0.000004^{***}
Claude: B vs RAG	ISO exact match	McNemar	1.0 (ns)
Claude: B vs RAG	Coverage (relaxed)	McNemar	0.057 (ns)
RQ2: Model Comparison			
Baseline: GPT vs Claude	ISO exact match	McNemar	$< 0.000001^{***}$
RAG: GPT-5.1 vs Claude	ISO exact match	McNemar	0.125 (ns)
RQ3: Mitigation Quality			
GPT-5.1: B vs RAG	Semantic recall	Wilcoxon	0.935 (ns)
GPT-5.1: B vs RAG	ROUGE-L F1	Wilcoxon	0.000020^{***}
GPT-5.1: B vs RAG	LLM-judge pairwise	Binomial	80.7% RAG wins
B vs RAG	ENISA quality (expert)	Mann-Whitney	$< 0.001^{***}$
B vs RAG	ISO quality (expert)	Mann-Whitney	0.396 (ns)

5.7 Summary of Findings

The key findings from this experiment are summarised below.

RQ1: Clause identification and coverage assessment

Retrieval-augmented grounding significantly improves clause identification for GPT-5.1, increasing ISO accuracy from 43.0% to 79.6%. The improvement is driven by the retrieval step, which surfaces the correct clause in the top-5 candidates in 95.7% of cases. For ENISA, RAG enables clause identification that is not possible from parametric knowledge alone. Coverage assessment accuracy improves from 78.5% to 98.9% for GPT-5.1 under relaxed evaluation. For Claude Opus 4.6, which already achieves strong ISO accuracy (86.0%) from parametric knowledge, RAG does not provide a statistically significant improvement in ISO clause prediction. However, RAG remains necessary for ENISA identification regardless of the model.

RQ2: Effect of LLM model choice

The choice of LLM model significantly affects baseline performance, with Claude demonstrating stronger parametric knowledge of ISO 27002 than GPT-5.1. Under RAG, this gap narrows to a non-significant 5.3 percentage points. Retrieval performance is model-independent. This suggests that RAG reduces dependency on model-specific domain knowledge. Whether Claude's stronger baseline performance stems from architectural differences, training data composition, or post-training alignment cannot be determined from external evaluation alone—the pre-training corpora of commercial LLMs are not publicly disclosed. What is empirically observable is that Claude's parametric knowledge of ISO 27002 clause structure is more accurate, regardless of the underlying cause. Under RAG, this advantage is neutralised because both models receive the same retrieved context.

RQ3: Mitigation recommendation quality

Three evaluation methods yield partially divergent results for ISO outputs. NLP metrics show slight baseline advantage due to lexical divergence; LLM-as-Judge evaluation shows RAG produces higher-quality recommendations (80.7% pairwise win rate); expert evaluation shows no significant difference due to a ceiling effect. For ENISA outputs, all methods converge: RAG is clearly superior. The disagreement between evaluation methods for ISO outputs is itself a methodological finding—it demonstrates that lexical overlap metrics are insufficient for assessing generative compliance outputs and that multi-method evaluation is necessary.

6

Discussion

This chapter discusses the implications of the findings, situates the results within the context of prior work, and identifies limitations and delimitations of the study.

6.1 Interpretation of Results

The results demonstrate that the effect of retrieval-augmented grounding depends on the model's existing parametric knowledge of the regulatory domain. For models with weaker domain-specific knowledge, RAG substantially improves clause identification accuracy. For models that already possess strong parametric knowledge, the benefit of RAG shifts from accuracy improvement to enabling access to regulatory frameworks absent from pre-training data and providing traceable evidence chains. This distinction has practical implications: organisations may choose between investing in RAG infrastructure or selecting models with stronger parametric knowledge, depending on their traceability and auditability requirements.

The finding that RAG narrows the performance gap between models aligns with the practical needs identified by Esposito et al. [15], who emphasise that AI systems in governance contexts must be transparent and defensible. By anchoring each compliance finding in retrieved regulatory text, RAG provides a mechanism for traceability that is independent of the model's internal knowledge—each clause selection is supported by explicit evidence rather than implicit parametric associations. This has implications for regulatory contexts where findings must be auditable and defensible.

The observation that both models fail completely on ENISA baseline (0% accuracy) while achieving moderate accuracy under RAG highlights the importance of retrieval for newer or less widely-known regulatory frameworks. However, the moderate ENISA accuracy under RAG (33–35%) also indicates that retrieval alone is not sufficient—improvements in the generation step (e.g., better reasoning over retrieved candidates) may be needed. This aligns with the finding by Niu et al. [50] that RAG-based compliance approaches require careful tuning of both retrieval and generation components to achieve high accuracy.

6.2 Mitigation Quality and Evaluation Challenges

The divergence between NLP metrics, LLM-as-Judge evaluation, and expert ratings reveals fundamental limitations of current evaluation approaches for generative compliance outputs. Surface-level metrics penalise RAG for producing verbose, framework-aligned text that uses different vocabulary from the reference phrasing, while LLM-as-Judge evaluation captures practical quality advantages that lexical overlap cannot measure.

The expert evaluation reveals a ceiling effect for ISO outputs: both pipelines meet the practitioner's quality threshold for ISO-grounded mitigations. This suggests that for well-known regulatory frameworks, both grounded and ungrounded LLMs can produce recommendations that are practically useful. The critical differentiator is output quality for frameworks not well-represented in pre-training data, where RAG enables meaningful generation that the baseline cannot produce.

This finding has methodological implications for future evaluations of compliance automation systems. Lexical overlap metrics alone are insufficient, particularly when systems produce framework-aligned text that is correct but phrased differently from reference standards. The disagreement between methods is itself a contribution: it demonstrates that multi-method evaluation combining automated scoring, LLM-as-Judge, and expert assessment provides a more complete and reliable picture of system quality than any single method alone.

6.3 Practical Implications

The results have several practical implications for organisations seeking to automate regulatory gap analysis. First, retrieval infrastructure is particularly important for regulatory frameworks that are not well-represented in LLM pre-training data. Both models failed completely on ENISA baseline, demonstrating that parametric knowledge cannot be relied upon for newer or less widely-disseminated frameworks. For established frameworks like ISO 27002, models with strong parametric knowledge may produce adequate results without retrieval, though traceability is lost.

Second, the high retrieval recall suggests that the retrieval component is effective at surfacing relevant clauses. The gap between retrieval performance and final selection accuracy indicates that improvements in the generation step could yield further gains without changes to the retrieval infrastructure.

Third, the conservative bias observed in coverage assessment—where both pipelines rarely predict full coverage—represents a cautious failure mode that may be acceptable in compliance contexts where overlooking a gap is more costly than flagging a non-existent one. Practitioners should be aware of this tendency when interpreting pipeline outputs.

6.4 Limitations

Several factors may affect the generalisability or interpretation of the results.

The evaluation relies on comparing LLM outputs against the Golden Standard Reference (GSR), which was developed with expert input but may contain inherent subjectivity. Any errors in the GSR directly affect the reported accuracy metrics. To mitigate this concern, the expert evaluation validates a subset of the automated findings, achieving near-perfect agreement (Cohen's $\kappa \geq 0.97$), which provides confidence that the GSR accurately captures correct clause mappings.

Large Language Models are probabilistic by nature. While temperature was set to zero to minimise non-determinism, minor stochastic variation may still occur between runs. The experiment uses single-run execution; repeated runs with confidence intervals are left for future work.

Regulatory compliance in the information security domain is often open to interpretation. The boundary between "partially covered" and "fully covered" involves subjective judgment that both the GSR construction and the pipeline's coverage assessment must navigate. The relaxed accuracy metric and Spearman's rank correlation partially address this by accepting adjacent predictions as correct.

The experiment results are tied to specific LLM model versions (GPT-5.1 and Claude Opus 4.6). Given the rapid evolution of LLM capabilities, findings regarding parametric knowledge strength and hallucination rates may change with newer model iterations. However, the relative comparison between baseline and RAG configurations is expected to remain informative, as the underlying mechanism (grounding vs. parametric knowledge) is model-architecture-independent.

The mitigation quality evaluation is based on a single rater (Expert B), which precludes direct inter-rater reliability analysis for recommendation quality. However, for traceability evaluation, indirect inter-rater agreement is achieved through the separation of roles: Expert A defined correct clause mappings (via the GSR) and Expert B independently confirmed them ($\kappa \geq 0.97$). The mitigation quality ratings nonetheless represent one professional's perspective and may not generalise to all compliance practitioners.

Use of licensed ISO/IEC 27001 text restricts full public release of the regulatory index used for retrieval, limiting external replication of the retrieval component. The prompts, evaluation scripts, and experimental methodology are fully documented.

Regarding generalisability of the input data: the input format used in this study—structured risk descriptions with optional mitigation and control fields—is representative of standard GRC platforms and risk register formats used across industries. The pipeline requires only free-text risk descriptions and optional mitigation text, which are available in any organisation maintaining a risk register or ISMS documentation. The regulatory indexes (ISO/IEC 27002 and ENISA NIS2 Tech-

nical Implementation Guidance) are publicly available standards. Therefore, the approach is replicable by any organisation with access to an LLM API and their existing risk management documentation, without requiring proprietary data formats or specialised tooling.

6.5 Delimitations

This study is intentionally narrowed in scope to ensure feasibility and depth of analysis.

The study focuses exclusively on inference-time grounding techniques. Fine-tuning and post-training are excluded from the evaluation, as the goal is to assess scalable and cost-effective techniques that do not require model modification. This means the results do not speak to whether fine-tuned models could achieve superior performance.

The empirical evaluation is restricted to the information security domain, specifically the 93 controls defined in ISO/IEC 27002 (chapters 5–8) and corresponding ENISA NIS2 clauses. While the underlying techniques may be applicable to other regulatory frameworks (e.g., GDPR, sustainability regulations), this study does not validate performance across multiple domains.

The analysis covers all four ISO 27002 control themes (organisational, people, physical, technological) and evaluates against specific ENISA NIS2 chapters. While this provides broad coverage of the ISO standard, the ENISA evaluation is limited to the chapters represented in the dataset.

The analysis is restricted to textual regulatory documents and organisational artifacts. Complex visual data (e.g., architectural diagrams or scanned documents) is considered out of scope unless pre-processed into text.

7

Conclusion

This thesis evaluated whether Retrieval-Augmented Generation (RAG) improves the accuracy, recommendation quality, and clause-level traceability of LLM-assisted regulatory gap analysis in the information security domain. A controlled computational experiment compared baseline (non-grounded) and RAG configurations for two LLM models (GPT-5.1 and Claude Opus 4.6) on a dataset of 93 compliance risks evaluated against an expert-validated Golden Standard Reference.

The findings show that retrieval-augmented grounding significantly improves regulatory clause identification for models with weaker parametric domain knowledge. For GPT-5.1, ISO clause prediction accuracy improved from 43.0% to 79.6%, and ENISA identification was enabled where it was previously impossible. Coverage assessment accuracy reached 98.9% under relaxed evaluation. For Claude Opus 4.6, which already achieves strong ISO accuracy from parametric knowledge, RAG did not provide a statistically significant improvement for ISO clauses but remained necessary for ENISA identification.

The choice of LLM model significantly affects baseline performance, but under RAG the inter-model gap narrows from 43 percentage points to a non-significant 5.3 points. This suggests that retrieval-augmented grounding reduces dependency on model-specific domain knowledge and can serve as an equaliser across architectures.

Mitigation quality evaluation revealed a methodological finding: NLP metrics, LLM-as-Judge, and expert evaluation yield divergent results for ISO outputs due to their different measurement properties. For ENISA outputs, all methods agree that RAG produces higher-quality content. The disagreement for ISO outputs demonstrates that lexical overlap metrics are insufficient for evaluating generative compliance outputs and that multi-method evaluation is necessary.

Expert validation achieved near-perfect agreement with the automated evaluation methodology (Cohen's $\kappa \geq 0.97$), confirming that automated exact-match evaluation accurately reflects domain expert judgment.

7.1 Contributions

This thesis makes the following contributions:

1. Empirical evidence that RAG significantly improves regulatory clause identification for LLMs with weaker parametric domain knowledge, while providing limited benefit for models with strong existing knowledge of the regulatory framework.

2. Demonstration that RAG acts as a model equaliser, reducing dependency on model-specific domain knowledge for regulatory gap analysis.

3. Evidence that RAG is essential for identifying clauses from regulatory frameworks not represented in LLM pre-training data (ENISA NIS2).

4. A methodological finding that traditional NLP metrics (ROUGE, BERTScore) are insufficient for evaluating generative compliance outputs and that multi-method evaluation is necessary.

5. A validated evaluation methodology combining automated exact-match evaluation, LLM-as-Judge, and expert assessment for regulatory gap analysis systems.

7.2 Future Work

Several directions for future research emerge from this work. First, evaluating agentic architectures (multi-step reasoning with tool use) as an alternative or complement to RAG for regulatory compliance tasks. Second, extending the evaluation to additional regulatory frameworks beyond ISO/IEC 27002 and ENISA NIS2 to assess generalisability. Third, conducting repeated experimental runs to quantify LLM non-determinism and compute confidence intervals. Fourth, evaluating with multiple domain experts to enable inter-rater reliability analysis. Fifth, investigating retrieval improvements (e.g., re-ranking, query expansion) to close the gap between top-5 recall and final selection accuracy, particularly for ENISA clauses.

7.3 Disclosure of AI Use

Large Language Models were used as tools during the preparation of this thesis in the following ways:

- **Pipeline implementation:** LLMs (GPT-5.1 and Claude Opus 4.6) were used as the experimental objects being evaluated—they are the subject of the study, not merely tools for writing.

- **Writing assistance:** An LLM was used to assist with drafting, restructuring, and revising thesis text. All content was reviewed, verified, and edited by the author. The research design, analysis, interpretation, and conclusions are the author's own work.

- **Code development:** LLM-assisted code generation was used for evaluation

scripts and data processing pipelines. All code was reviewed and tested by the author.

The evaluation scripts and supplementary materials are available in the project repository.[1]

[1] Repository URL to be added upon publication.

Bibliography

[1] A. Tarantino, *Governance, Risk, and Compliance Handbook: Technology, Finance, Environmental, and International Guidance and Best Practices*. Hoboken, NJ, USA: Wiley, 2008. doi: 10.1002/9781118269145.

[2] *Guidelines for auditing management systems*, ISO Standard 19011:2018, Jul. 2018.

[3] S. Romanosky, "Examining the costs and causes of cyber incidents," *Journal of Cybersecurity*, vol. 2, no. 2, pp. 121–135, Dec. 2016. doi: 10.1093/cybsec/tyw001.

[4] N. Racz, E. Weippl, and A. Seufert, "A frame of reference for research of integrated governance, risk and compliance (GRC)," in *Trust and Privacy in Digital Business* (Lecture Notes in Computer Science, vol. 6264), S. Fischer-Hübner *et al.*, Eds. Berlin, Heidelberg: Springer, 2010, pp. 106–117. doi: 10.1007/978-3-642-15152-1_10.

[5] K. Spanaki and A. Papazafeiropoulou, "Analysing the governance, risk and compliance (GRC) implementation process: Primary insights," in *Proc. 21st European Conf. Information Systems (ECIS)*, Utrecht, Netherlands, 2013, Paper 24. [Online]. Available: https://aisel.aisnet.org/ecis2013_cr/24.

[6] N. Racz, E. Weippl, and A. Seufert, "Integrating IT governance, risk, and compliance management processes," in *Databases and Information Systems VI* (Frontiers in Artificial Intelligence and Applications, vol. 224), J. Barzdins and M. Kirikova, Eds. Amsterdam, The Netherlands: IOS Press, 2011, pp. 325–338. doi: 10.3233/978-1-60750-687-4-325.

[7] *Information security, cybersecurity and privacy protection — Information security management systems — Requirements*, ISO/IEC 27001:2022, International Organization for Standardization, Geneva, Switzerland, 2022.

[8] *Information security, cybersecurity and privacy protection — Information security controls*, ISO/IEC 27002:2022, International Organization for Standardization, Geneva, Switzerland, 2022.

[9] Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union (NIS2 Directive), *Official Journal of the European Union*, L 333, pp. 80–152, Dec. 2022.

[10] European Union Agency for Cybersecurity (ENISA), "Technical implementation guidance on the cybersecurity measures for the implementing regulation under NIS2," version 1.0, Jun. 2025. [Online]. Available: https://www.enisa.europa.eu.

[11] M. Antunes, M. Maximiano, R. Gomes, "A Customizable Web Platform to Manage Standards Compliance of Information Security and Cybersecurity Auditing" in *Procedia Computer Science*, 2022, vol. 196, pp. 36–43. doi: 10.1016/j.procs.2021.11.070.

[12] L. Axon, A. Erola, A. Rensburg, J. Nurse, M. Goldsmith, S. Creese "Practitioners' Views on Cybersecurity Control Adoption and Effectiveness" in *Proceedings of the 16th International Conference on Availability, Reliability and Security*, Aug. 2021, pp. 1–10. doi: 10.1145/3465481.3470038.

[13] Z. Li, X. Wang, Q. Zhang, *et al.*, "Evaluating the quality of large language model-generated cybersecurity advice in GRC settings," Procedia Computer Science, Jun. 2024. doi: 10.21203/rs.3.rs-4608321/v1.

[14] A. Berger *et al.*, "Towards automated regulatory compliance verification in financial auditing with large language models," in *2023 IEEE Int. Conf. Big Data (BigData)*, Sorrento, Italy, 2023, pp. 4626–4635. doi: 10.1109/BigData59044.2023.10386518.

[15] M. Esposito, F. Palagiano, V. Lenarduzzi, and D. Taibi, "On large language models in mission-critical IT governance: Are we ready yet?" in *2025 IEEE/ACM 47th Int. Conf. Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, Ottawa, ON, Canada, 2025, pp. 504–515. doi: 10.1109/ICSE-SEIP66354.2025.00050.

[16] D. Seifert, L. Jöckel, A. Trendowicz, M. Ciolkowski, T. Honroth, and A. Jedlitschka, "Can large language models (LLMs) compete with human requirements reviewers? Replication of an inspection experiment on requirements documents," in *Product-Focused Software Process Improvement (PROFES 2024)* (Lecture Notes in Computer Science, vol. 15452), D. Pfahl *et al.*, Eds. Cham: Springer, 2024. doi: 10.1007/978-3-031-78386-9_3.

[17] H. Cheng, J. Husen, S. Peralta, B. Jiang, N. Yoshioka, N. Ubayashi, and H. Washizaki, "Generative AI for requirements engineering: A systematic literature review," arXiv preprint arXiv:2409.06741, 2024. doi: 10.48550/arXiv.2409.06741.

[18] S. Hassani, "Enhancing legal compliance and regulation analysis with large

language models," in *Proc. IEEE 32nd Int. Requirements Eng. Conf. (RE)*, 2024, pp. 507–511. doi: 10.1109/RE59067.2024.00065.

[19] A. Salman, S. Creese, and M. Goldsmith, "Position paper: Leveraging large language models for cybersecurity compliance," in *2024 IEEE European Symp. Security and Privacy Workshops (EuroS&PW)*, Vienna, Austria, 2024, pp. 496–503. doi: 10.1109/EuroSPW61312.2024.00061.

[20] Y. Wang, Y. Cai, M. Chen, Y. Liang, and B. Hooi, "Primacy effect of ChatGPT," arXiv preprint arXiv:2310.13206, 2023.

[21] T. Lu, M. Gao, K. Yu, A. Byerly, and D. Khashabi, "Insights into LLM long-context failures: When transformers know but don't tell," arXiv preprint arXiv:2406.14673, 2024. doi: 10.48550/arXiv.2406.14673.

[22] Y. Du, M. Tian, S. Ronanki, S. Rongali, S. Bodapati, A. Galstyan, A. Wells, R. Schwartz, E. A. Huerta, and H. Peng, "Context length alone hurts LLM performance despite perfect retrieval," in *Findings of the Assoc. for Computational Linguistics: EMNLP*, 2025, pp. 23281–23298. doi: 10.48550/arXiv.2510.05381.

[23] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, 2020, pp. 9459–9474. doi: 10.48550/arXiv.2005.11401.

[24] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009. doi: 10.1561/1500000019.

[25] S. Ganatra and P. Bhattacharyya, "Retrieval-augmented generation: A comprehensive survey on bridging language models and external knowledge," Center for Indian Language Technology (CFILT), IIT Bombay, Mumbai, India, Tech. Rep., 2025.

[26] A. V. Kozhipuram, S. Shailendra, and R. Kadel, "Retrieval-augmented generation vs. baseline LLMs: A multi-metric evaluation for knowledge-intensive content," *Information*, vol. 16, no. 9, Art. no. 766, 2025.

[27] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Cham, Switzerland: Springer, 2012.

[28] S. Es, J. James, L. Espinosa Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval-augmented generation," in *Proc. EACL: System Demonstrations*, 2024.

[29] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging LLM-as-a-

judge with MT-bench and Chatbot Arena," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[30] M. Ivgi, U. Shaham, and J. Berant, "Efficient long-text understanding with short-text models," *Trans. Assoc. Comput. Linguist.*, vol. 11, pp. 284–299, 2023.

[31] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 157–173, 2024.

[32] J. H. Chin, P. Zhang, Y. X. Cheong, and J. Pan, "Automating security audit using large language model based agent: An exploration experiment," arXiv preprint arXiv:2505.10732, 2025. doi: 10.48550/arXiv.2505.10732.

[33] J. Hassine, "An LLM-based approach to recover traceability links between security requirements and goal models," in *Proc. 28th Int. Conf. Evaluation and Assessment in Software Engineering (EASE)*, 2024, pp. 643–651. doi: 10.1145/3661167.3661261.

[34] A. D. Rodriguez, K. R. Dearstyne, and J. Cleland-Huang, "Prompts matter: Insights and strategies for prompt engineering in automated software traceability," arXiv preprint arXiv:2308.00229, 2023.

[35] M. Usman, M. Felderer, M. Unterkalmsteiner, E. Klotins, D. Mendez, and E. Alégroth, "Compliance requirements in large-scale software development: An industrial case study," in *Product-Focused Software Process Improvement (PROFES 2020)* (Lecture Notes in Computer Science, vol. 12562), M. Morisio, M. Torchiano, and A. Jedlitschka, Eds. Cham, Switzerland: Springer, 2020, pp. 385–401.

[36] W. Gan, "Large language models empowering compliance checks and report generation in auditing," *World J. Inf. Technol.*, vol. 2, no. 2, pp. 35–39, 2024. doi: 10.61784/wjit3003.

[37] G. Kamradt, "Needle in a Haystack: Pressure testing LLMs," GitHub repository, 2023. [Online]. Available: https://github.com/gkamradt/LLMTest_NeedleInAHaystack. [Accessed: Jan. 16, 2026].

[38] I. Kazlaris, E. Antoniou, K. Diamantaras, and C. Bratsas, "From illusion to insight: A taxonomic survey of hallucination mitigation techniques in LLMs," *AI*, vol. 6, no. 10, Art. no. 260, 2025. doi: 10.3390/ai6100260.

[39] A.-H. Dang, V. Tran, and L.-M. Nguyen, "Survey and analysis of hallucinations in large language models: Attribution to prompting strategies or model behavior," *Frontiers in Artificial Intelligence*, vol. 8, Art. no. 1622292, 2025. doi: 10.3389/frai.2025.1622292.

[40] OpenAI, "GPT-4o system card," 2024. [Online]. Available: https://openai.com/index/gpt-4o-system-card/. [Accessed: Apr. 10, 2026].

[41] OpenAI, "New embedding models and API updates," Jan. 2024. [Online]. Available: https://openai.com/index/new-embedding-models-and-api-updates/. [Accessed: Apr. 10, 2026].

[42] Microsoft, "Azure AI Search documentation," 2024. [Online]. Available: https://learn.microsoft.com/en-us/azure/search/. [Accessed: Apr. 10, 2026].

[43] T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2023.

[44] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with ChatGPT," arXiv preprint arXiv:2302.11382, 2023.

[45] S. Das, N. Deb, N. Chaki, and A. Cortesi, "A multi-agent RAG framework for regulatory compliance checking of software requirements," *ACM Trans. Softw. Eng. Methodol.*, 2025 (to appear). doi: 10.1145/3785472.

[46] A. Masoudifard, M. Mowlavi Sorond, M. Madadi, M. Sabokrou, and E. Habibi, "Leveraging Graph-RAG and prompt engineering to enhance LLM-based automated requirement traceability and compliance checks," arXiv preprint arXiv:2412.08593, 2024.

[47] J. Sun, Z. Luo, and Y. Li, "A compliance checking framework based on retrieval augmented generation," in *Proc. 31st Int. Conf. Computational Linguistics (COLING)*, 2025, pp. 2603–2615.

[48] Y. Han, A. Ceross, and J. H. M. Bergmann, "Standard applicability judgment and cross-jurisdictional reasoning: A RAG-based framework for medical device compliance," arXiv preprint arXiv:2506.18511, 2025.

[49] B. V. Balu, F. Geissler, F. Carella, J.-V. Zacchi, J. Jiru, N. Mata, and R. Stolle, "Towards automated safety requirements derivation using agent-based RAG," in *Proc. AAAI-MAKE Spring Symposium*, 2025. arXiv:2504.11243.

[50] F. Niu, R. Pan, L. C. Briand, H. Hu, and K. Koravadi, "TVR: Automotive system requirement traceability validation and recovery through retrieval-augmented generation," arXiv preprint arXiv:2504.15427, 2025.

[51] C. Arora, F. Wang, C. Tantithamthavorn, A. Aleti, and S. Kenyon, "From domain documents to requirements: Retrieval-augmented generation in the space industry," in *2025 IEEE 33rd Int. Requirements Eng. Conf. (RE)*, 2025, pp. 303–307. doi: 10.1109/RE63999.2025.00036.

[52] M. Khalid and Y. Uygun, "An industrial-scale retrieval-augmented generation framework for requirements engineering: Empirical evaluation with automotive manufacturing data," arXiv preprint arXiv:2603.20534, 2026.

[53] S. Allani, K. Bou-Chaaya, and H. Rais, "ELISAR: A multi-agent cybersecurity framework integrating retrieval-augmented generation for Blue, Red, and GRC operations," *World Wide Web*, vol. 28, 2026. doi: 10.1007/s11280-025-01393-5.

[54] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP-IJCNLP*, 2019, pp. 3982–3992.

[55] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating text generation with BERT," in *Proc. ICLR*, 2020.

[56] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain, 2004, pp. 74–81.

[57] T. D. Cook and D. T. Campbell, *Quasi-Experimentation: Design and Analysis Issues for Field Settings*. Boston, MA, USA: Houghton Mifflin, 1979.

[58] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG evaluation using GPT-4 with better human alignment," in *Proc. EMNLP*, 2023, pp. 2511–2522.

[59] Cook, T.D. and Campbell, D.T. (1979) *Quasi-Experimentation: Design and Analysis Issues for Field Settings*. Houghton Mifflin, Boston, MA.

[60] Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R. and Zhu, C. (2023) G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2511–2522.

[61] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-tuning or retrieval? Comparing knowledge injection in LLMs," arXiv preprint arXiv:2312.05934, 2024.

[62] Y. Gao, Y. Xiong, M. Wang, and H. Wang, "Modular RAG: Transforming RAG systems into LEGO-like reconfigurable frameworks," arXiv preprint arXiv:2407.21059, 2024.

A

Prompt Listings and Iteration History

The complete prompt listings, iteration history, and supplementary implementation material are available in the accompanying GitHub repository:

`https://github.com/jena94/masters-thesis-rag-compliance`