Investigating the use of Large Language Models in software security requirements: Results of a literature review

Amália Melo
LabES, ICMC
University of São Paulo.
São Carlos - SP, Brazil
amalia.melo@usp.br

Lucas Almeida
LabES, ICMC
University of São Paulo.
São Carlos - SP, Brazil
almeidalucas@usp.br

Lina Garcés
LabES, ICMC
University of São Paulo.
São Carlos - SP, Brazil
linagarces@usp.br

ABSTRACT

This study investigates the application of Large Language Models (LLMs) in the context of security requirements engineering through the conduction of a rapid literature review. The review enabled the characterization of current research in this domain with respect to: (i) the purposes for which LLMs are employed in security requirements activities; (ii) the families of LLMs explored (e.g., GPT, BERT, LLaMA), their capabilities (e.g., classification, generation), and underlying architectures (e.g., encoder, decoder, encoder-decoder); (iii) the techniques adopted for conditioning or guiding LLM behavior; (iv) the datasets used to train, fine-tune, or validate these models; and (v) the evaluation metrics applied to assess the performance of LLMs in supporting security requirements tasks. The findings contribute to a structured understanding of the current state of research and highlight key trends, gaps, and opportunities for advancing the use of LLMs in secure software engineering.

KEYWORDS

design patterns, software engineering education, jigsaw, active learning

1 Introduction

The security of a software system is defined as the ability to protect information and data, according to authorization rules, security standards, and data privacy and protection regulations. In addition, security aims to defend software against attack patterns by malicious agents [22]. For this, data must be protected during storage, presentation to users, and transmission over communication networks [21].

Security-by-design is a transversal area of software systems that aims to integrate security activities into all aspects of software development to minimize vulnerabilities that can be exploited during software operation. For this, security concerns are starting to be considered in the early stages of software development, such as analysis and requirements engineering, namely, analysis, elicitation, specification, validation, traceability, change management, among others.

In another perspective, according to [26], the application of Large Language Models (LLMs) in software engineering can be widely explored, ranging from requirements analysis in the early stages of the development lifecycle to the generation of code, test cases, or infrastructure as a code [14, 20, 33]. LLMs are computational models that represent human language using statistical structures and semantic relationships across an extensive textual database [15]. There is a vast variety of models proposed by academics and by the industry, for instance, BERT, Llama, GPT, Gemini, among others. LLM's architecture can be of types such as encode (for classification purposes), decode (for generation purposes), or encode-decode (aiming classification and generation) [14, 20].

Regarding the techniques for using LLMs, it is possible to enumerate transfer learning, fine-tuning, retrieval augmentation generation (RAG), and prompt engineering, among others [7, 9, 23, 24]. Unlike techniques that require (partially) retraining the LLM (e.g., fine-tuning, RAG), prompt engineering, also known as in-context learning, refers to methods used to steer the behavior of LLMs through carefully designed prompts, without modifying the model weights.

As observed in related secondary studies on LLMs for software engineering [14, 20, 32], those models have been applied across different phases of requirements engineering, such as: (i) **Requirements Elicitation:** LLMs can support in automatically generating requirements, creating interview scripts, and extracting relevant information from documents; (ii) **Requirements Classification:** The LLMs can outperform traditional machine learning (ML) and natural language processing (NLP) approaches, offering generalization, automation, and adaptability to complex tasks and domains, reducing the need for manual processes; (iii) **Ambiguity and Completeness Management:** While challenges remain in these tasks, the use of LLMs, particularly those based on BERT, has brought significant improvements. Notably, the positive results of using techniques such as prompt-based learning with few-shot learning and the combination of traditional methods with NLP techniques are promising; and (iv) **Requirements Traceability:** The application of LLMs alongside techniques such as knowledge distillation, multi-task learning, and semi-supervised learning have offered considerable improvements in the scalability and efficiency of the traceability process, even in large and complex software projects.

Given the benefits identified in the state-of-the-art literature on the application of LLMs in the broader domain of requirements engineering, this study aims to explore the extent to which similar advantages can be observed in the context of security requirements engineering. To this end, we conducted a rapid literature review [10] to identify and characterize primary studies that apply or propose the use of LLMs to support activities related to security requirements.

The remainder of this paper is structured as follows: Section 2 describes the methodological design of this secondary study. Section 3 presents the findings of the literature review. Finally, Section 4 provides concluding remarks and directions for future work.

2 Methods

This study aims to offer an overview on the use of LLMs for security software requirements and characterize those studies regarding the following research questions (RQs):

- **RQ1 -** What are the purposes, related to security requirements, that motivated the studies to use LLM?
- **RQ2 -** What LLM models have been used for security software requirements?
- **RQ3 -** Which LLM techniques (e.g., RAG, fine-tuning, prompt engineering) are being used by primary studies?
- **RQ4 -** Which databases are being used in research of LLM-based security software requirements?
- **RQ5 -** How the research of LLM-based security software requirements are being evaluated and what metrics are being used for it?

For this purpose, the rapid literature review (RLR) method was conducted following the guidelines in [10]. Figure 1 shows the process of this review. Each stage and its results are explained as follows.

Stage 1 - Primary studies search. The searching was done by one research in May, 2025. The search strategy contemplated two moments. Firstly, the following search string was running in Scopus database[1]: "***TITLE-ABS-KEY*** *(("requirement" OR "requirements" OR "requirement specification" OR "requirement engineering")* ***AND*** *("GenAI" OR "LLM" OR "generative artificial intelligence" OR "generative AI" OR "language model")* ***AND*** *("security" OR "data protection" OR "data privacy")* ***AND*** *(software OR "software engineering")´ ´*. This first searching returned 26 studies. Following, those 26 studies were uploaded in Research Rabbit [2], an AI-based tool that discovers similar work based on snowballing (backward and forward) techniques. The Research Rabbit returned 51 studies that matched in similarity to those returned by Scopus. As result of this first stage we obtained 77 studies.

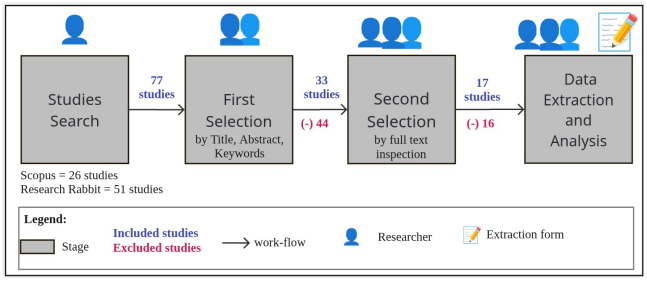

Figure 1: Literature review process

Stage 2 - First selection. This selection was done by two researchers whom read the title abstract, and keywords of the 77 studies. To be selected for the next step, studies must accomplish the following inclusion criteria: **IC1 -** The study propose or uses LLM models to solve a problem related with security software requirements. Studies were excluded when they met at least one of the following exclusion criteria: **EC1 -** The study solves a problem related with security software requirements, however it doesn't use LLM models; **EC2 -** The study is written in a language different from English, Portuguese, or Spanish; **EC3 -** The full-text of the study is not available for downloading; **EC4 -** The study was published as a short paper, thesis, monograph, or technical report. As a result of this stage, 44 studies were excluded and 33 were included for the next step.

Stage 3 - Second Selection. The full-text of the 33 studies was read and the inclusion and exclusion criteria were applied. This activity was conducted by two researchers. A third researcher intervened in cases where there were disagreements between the other researchers. As a result, 16 studies were excluded and 17 were included as the final set of primary studies of this literature review. The included studies are listed in Table 1.

Stage 4 - Data extraction and analysis. For data extraction was proposed a Google Forms containing all information required to answer the research questions. Two reviewers selected the articles and extracted the data. A third, more experienced reviewer participated in resolving disagreements or questions. Quality was assessed based on the methodological rigor of the primary studies.The final set of primary studies and the extracted data from them are available in [13]. Furthermore, the extracted data was analyzed using qualitative and narrative synthesis methods, as recommended by [29].

3 Results

This study was conducted between November 2024 and July 2025. Accordingly, the final set of primary studies, presented in Table 1, includes works published up to June 2025. The table also provides the extracted data from each of the selected studies.

A significant portion of the studies were published within the last 18 months, with 41% (7 out of 17) published in 2024 and 29.5% (5 out of 17) in 2025. The remaining 29.5% (5 out of 17) were published between 2020 and 2023.

The findings addressing the research questions are presented in the subsequent sections.

3.1 RQ1 - Purposes of using LLMs in security software requirements

As depicted in Figure 2, primary studies have researching the use of LLMs in security software requirements for a quite variety of purposes, namely, analysis (2/17), elicitation (2/17), specification (12/17), validation (2/17), traceability and change management (1/17), and documentation (3/17).

Our analysis reveals that the specification of security requirements has been the primary focus in the majority of the studies reviewed. Specifically, 70.5% (12 out of 17) of the primary studies report contributions in this area. Within this subset, requirements classification emerges as the most commonly cited motivation by 65% (11 out of 17) of the studies. These studies aimed to classify sets of software requirements into: (i) functional and non functional requirements; (ii) non functional requirements types, such as performance, usability, security, and availability; (iii) security properties of privacy, protection, availability, confidentiality, integrity, authentication, and authorization; (iv) requirements related, or not, to security; (v) requirements prone, or not, to privacy; and

[1] http://www.scopus.com
[2] https://www.researchrabbit.ai/

Table 1: Final set of primary studies on LLM-based security software requirements.

ID	Ref.	Year	Purpose	LLM Model(s)	Conditioning Method	Evaluation Metrics	Dataset(s)
S01	[11]	2024	Binary and multi-label classification	BERT	Fine-tuning; ensemble techniques	Recall, F1-Score, Hold-out, AUC-ROC, Precision	AI-CRAS-Dataset (own dataset) + CISPE handbook
S02	[8]	2024	Extraction and classification, and assessment of coesion, clarity, and precision.	GPT	Few-Shot Prompting	Humans Evaluation, 5-point Likert scale	ERTMS L3 requirements
S03	[5]	2025	Generation NFRs from FRs	Claude, DeepSeek, Gemini, GPT, Grok, LLaMA, Qwen	Few-Shot Prompting and Role prompting	Accuracy, confusion matrix, human feedback	FR_NFR_dataset, extracted in part from PURE
S04	[1]	2024	Conflict resolution, completeness, and feasibility validation.	GPT	Role prompting	Human Feedback	Dalpiaz user-stories [12]
S05	[4]	2023	Classifation, specification generation	AllMini, Bert4RE, SBERT, SObert	Zero - shot learning	F1-Score, Precision, Recall	PROMISE + SecReq
S06	[6]	2024	Classification	BERT		Accuracy, F1-score	PROMISE + IREC 2017 Data-Challenge (adapted)
S07	[2]	2024	Extraction of quality concerns	BERT, DistilBERT, RoBERTa, XLNET	Fine-tuning	Accuracy, F1-score, Precision and Recall	Own dataset compiled from user stories and acceptance criteria obtained from various sources
S08	[31]	2021	Classification of security properties	BERT, DistilBERT, XLNet	Fine Tuning	Confusion matrix, Recall, F1-Score, K-Fold Cross-validation, Precision	Own dataset built from PURE + SecReq + Riaz's dataset
S09	[16]	2024	Traceability and change management	GPT	Zero-Shot Prompting, Role prompting	Recall, F1-Score, Human evaluation, precision	TGRL Specification of the GRL Model for Virtual Interior Design, developed by undergraduate students.
S10	[19]	2024	Alignment with the ISO 27001 standard	CISO-BERT, SBERT	Fine-tuning	HPOS@kl	Own dataset built from ISO 27001 norm + BSI IT-Grundschutz + MAP (a professional mapping between the two standards)
S11	[3]	2025	Classification	BERT, DistilBERT, RoBERTa, XLNet	Fine-tuning	Accuracy, F1-score, precision, confusion matrix and F1 weitgh	Own dataset built from academic resources and online sources
S12	[18]	2020	Classification	NoRBERT (BERT)	Fine-tuned	Recall, F1-Score, K-Fold Cross-validation, weighted average F1-score and precision	PROMISE + Dalpiaz [12]
S13	[34]	2024	Classification	BART, BERT, DistilBERT, GPT, RoBERTa, T5	Fine-tuning	Accuracy, Precision, F1-score and Recall	SecReq + PROMISE + PURE + Kaggle (REQ-Class) + IoTAC (adapted)
S14	[28]	2025	Classification	BERT, DistilBERT, RoBERTa, XLNet	Fine-tuning	Accuracy, Precision, Recall, F1-Score and confusion matrix	SecReq (adapted)
S15	[25]	2023	Classification	BERT, DistilBERT, DistilRoberta, Electra, XLNet	Prompting (few-shot) with examples	Acuraccy, confusion matrix, recall, F1-score, Hold-out and precision	Non Functional Requirements (own dataset)
S16	[27]	2022	Classification	BERT-MLM, NoRBERT, PRCBERT (BERT/RoBERTa), Trans_PRCBERT	Fine-tuning + vocab matching (DPV) + synonym/mention replacement	Precision, F1-score, Recall, Weighted-F1, paired t-testn, Cross-validation, K-Fold Cross-validation	NFR-SO (own dataset) + PROMISE + NFR-Review
S17	[17]	2024	Identify and classify privacy requirement	BERT, RoBERTa	Data augmentation, token separation, prompt learning?	F1-Score, Recall, Precision	Augmented PII dataset (privacy forums and vocab); single evaluation split Custom augmented set + DPV + Camper+ project; test set from Hadar et al. (2021)

(v) specifications related to any requirement (including security) or other kind of software concern.

Additionally, we identified that generative LLMs, specifically studies S02, S03, S04, S09, and S13, have been applied to various activities within the requirements engineering process. These applications include: conflict resolution during requirements analysis (S04); requirements elicitation, particularly the generation of security requirements based on functional requirements (S03); classification of requirements into functional (FR) and non-functional (NFR) categories, as well as sub-classification within NFRs (S13); validation of the completeness, cohesion, clarity, and precision of security requirements (S02 and S04); and support for traceability and change management of security requirements (S09). It is noteworthy that, for the latter two purposes, i.e, traceability and change management, only generative LLMs have been explored in the reviewed studies, differently from others activities where non generative LLMs also have been studied.

3.2 RQ2 - Models used for security requirements engineering

Twelve distinct LLMs families have been found across the 17 primary studies. Among the models with encoder-only architectures, BERT-based models (i.e., BERT-base/large, RoBERTa, DistilBERT, SBERT, and specialised variants such as CISO-BERT and Bert4RE) dominated, being used in 76% (13 out of 17) of studies (See Table 1). At the same time, ELECTRA appeared once (S15).

Regarding decoder-only families, GPT and XLNet were reported in 29,5% (5 out of 17) of studies, although for different purposes.

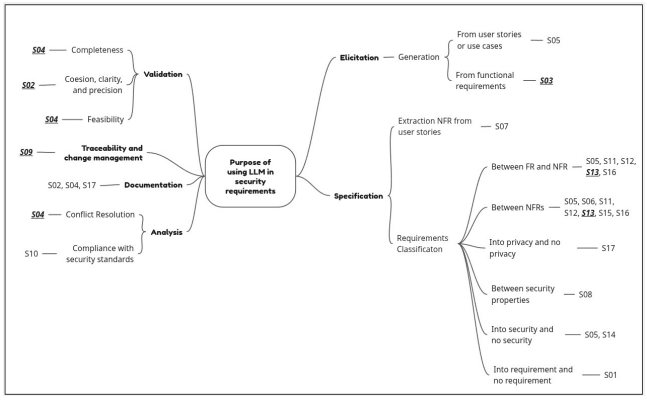

Figure 2: Classification of primary studies purposes.

XLNet first surfaced in 2021 (S08) and reappeared in 2023 (S15) with more force in last two years (S07, S11, S14 and S15); in every case, XLNet it was fine-tuned for text classification and evaluated alongside a BERT comparison, showing that early decoder-only models are still being treated as classifiers rather than generators. GPT variants (S02, S03, S04, S09, and S13), by contrast, were leveraged mainly for text generation, with the only exception being S13, in which the GPT-2 was used as a text classifier, alongside the only encoder–decoder models identified: BART and T5. A further cluster of decoder-only models, i.e., Claude, Gemini, DeepSeek, LLaMA, Grok, and Qwen, was benchmarked together in S03, the sole study that assesses eight distinct generators side by side.

23.4% (4 out of 17) of primary studies applied LLMs for generative purposes, all published from 2024 onward. In this scenario, S02 applied GPT-3.5 (ChatGPT) and GPT-4 (through Microsoft Copilot) to formalize railway cybersecurity requirements in the CNL4DSA controlled language; S04 evaluated GPT-3.5-turbo-16k (ChatGPT) on its capability to support professionals in assessing GDPR compliance within user stories; S09 employed GPT-3.5-turbo (ChatGPT) to generate security-related traceability links between natural-language requirements and GRL goal models; and S03 benchmarked eight different generators—GPT-4o-mini, Claude (claude-3-5-haiku/claude-3-7-sonnet), Gemini 1.5 pro, DeepSeek-V3, LLaMA-3.3, Grok-2 and Qwen-2.5, during automated generation of ISO/IEC 25010, aligned non-functional requirements from functional requirements.

3.3 RQ3 - Techniques used to engineer LLMs

LLMs can be subjected to pre-training, training, and post-training techniques, collectively referred to as methods for LLMs conditioning. Among them, as depicted in Figure 3, the most used were fine-tuning and prompt engineering variations, e.g., zero-shot, few-shot, and role-playing, some of them with hyperparameter settings. The fine-tuning technique has been present since the first analyzed year, 2020. It continues to be extensively studied over the subsequent five years in the domain of security requirements engineering, with approximately 58.82% (10 out of 17) of the studies applying this technique in some way. This technique was applied across different models with the primary goal of adapting them to the specific domain in which they were being used. The models used in these studies include the following: BERT (S01, S07, S08, S10 - S16), RoBERTa-base (S07, S11, S14, and S16), RoBERTa-large (S07 and S16), BERT-large (S07 and S16), DistilBERT (S07, S08, S14, S15), DistilRoBERTa (S15), SBERT (S10), CISO-BERT (S10), XLNet (S07, S11, S14, S15), Doc2Vec (S10), Electra-base and Electra-small (S15), and GPT, T5, and BART (S13).

Despite generative LLMs emerged more recently, prompt engineering techniques have are already being tested in research within the field of security requirements engineering. The models used include: GPT (S02, S04, and S09), and LLaMA, Anthropic Claude, Gemini, Grok, DeepSeek, and Qwen (S03). In general, prompt engineering techniques were applied in the phases of validation, analysis, and generation of security requirements. The hyperparameter tuning technique (i.e., setting temperature) was also used in combination with prompt engineering approaches in studies S03 and S09.

Four emerging methodologies were identified, i.e., zero-shot learning, few-shot learning, and prompt-learning, the first two applied in two studies each and the last one applied only once. The zero-shot learning (ZSL) technique, applied in S05 and S16, aims to perform learning tasks without using training data. The Few-Shot Learning (FSL) technique is conceptually similar to ZSL. Still, instead of using zero training data, it employs a small fraction (usually around 10%) during the training phase, with the remainder used for validation and testing. ZSL and FSL have been explored to generate security requirements from user stories and classify non-functional requirements. The prompt-learning technique involves adapting the model to transform a multiclass classification task into a binary classification task, e.g., using BERT combined with prompt variants (S16). Additionally, the stacking technique, traditionally used in machine learning, was applied in (S06) to compare the BERT model with other approaches such as GRU, LSTM, and RNN for the task of classifying non-functional requirements into their subcategories (S06).

We observed that the RAG technique has not been explored as a conditioning methodology for LLMs within the domain of security requirements engineering. However, it remains a promising technique to be investigated based on relevant results (e.g., reduction of hallucinations) observed in other research areas [9].

3.4 RQ4- Databases used for training and/or testing LLMs

The evidence indicates a limited variety of datasets employed across the reviewed studies. Notably, 29.4% (5 out of 17) of the studies utilize the PROMISE NFR dataset, either directly or indirectly, specifically, studies S05, S06, S11, S12, S13, and S16. The second most frequently used dataset is SecReq (utilized in 4 out of 17 studies: S05, S08, S13, S14), followed by PURE, which appears in 3 studies (S03, S08, S13). Both S04 and S17 made use of the Dalpiaz user stories dataset [12]. All other datasets appear only once, for instance, the AI-CRAS-Dataset in S01 or the IREC 2017 Data Challenge in S06.

These datasets are rarely applied in their original form; instead, many authors adapt or refine them to suit specific subdomains. Examples include the merging of PROMISE and IREC-2017 datasets in S06, the re-labeling of PROMISE in S12, and the creation of large hybrid datasets, such as the fusion of five different sources

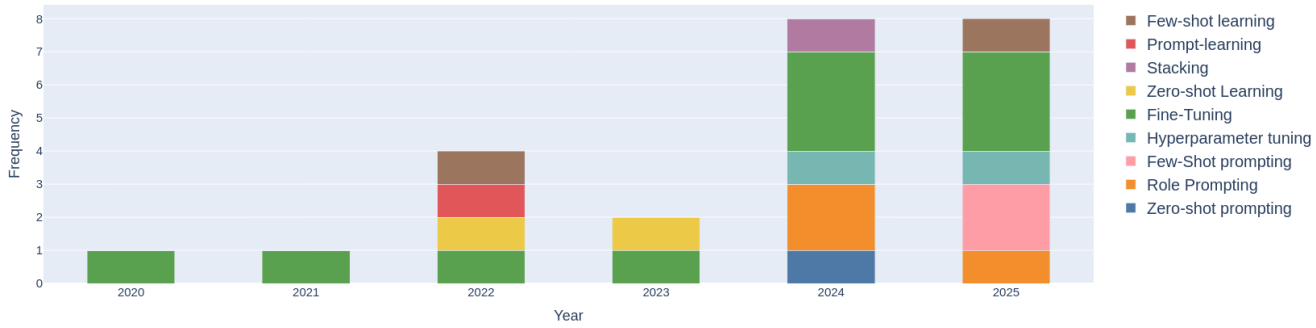

Figure 3: Conditioning techniques to guide LLMs' behavior.

(PROMISE, PURE, IoTAC, Kaggle (REQ-Class), and SecReq) in S13, accompanied by manual labeling of incomplete data.

The findings thus reveal a scarcity of publicly available datasets in the domain of security requirements. As a result, 59% (10 out of 17) of studies (S01, S03, S07, S08, S09, S10, S11, S13, S15, S17) were compelled to create their own datasets or heavily customize existing ones due to the lack of suitable public alternatives. Additionally, three studies (S13, S14, S17) reported the use of data augmentation techniques during pre-processing to expand the available data artificially.

Conversely, this data scarcity had less impact on studies employing generative AI approaches (S02, S03, S04, S09), as these did not involve fine-tuning the models. In such cases, datasets served primarily as sources of examples to support prompt engineering activities.

3.5 RQ5 - Evaluation of LLM-based security requirements proposals

When analyzing the metrics used by the articles to evaluate LLM models, a predominance of quantitative metrics over qualitative ones is observed. The main metrics applied are Precision, F1-Score, and Recall. This widespread use is believed to be related to the fact that these metrics are well-established in the machine learning literature and are particularly suitable for classification tasks, especially binary classification of requirements, which is a common focus in many of the analyzed studies.

Among the less frequently used metrics, HPOS@k [30] stands out. This is a precision-based metric applied in one study (S10) aimed at tracking and recommending security requirements. Another metric observed was the 5-Point Likert Scale, used in S02, where security requirements domain experts assessed the performance of prompt engineering techniques applied to requirements generation.

It was also noted that the evaluation of models using generative AI and prompt engineering techniques involved the participation of human experts (S02, S03, S04, and S09). In these cases, the output generated by the models underwent more detailed qualitative analysis by domain experts to assess aspects such as relevance, coherence, and adequacy of the generated requirements.

4 Final Remarks

Preliminary literature reviews [14, 20] identified an increasing application of LLMs to software engineering activities. However, the vast majority of research focuses on code generation, test case generation, and code inspection. In contrast, a significantly smaller portion of the literature addresses requirements engineering, and even fewer on security requirements engineering.

This imbalance highlights a clear need to enhance research efforts focused on automating critical security requirements engineering tasks, including analysis, specification, validation, traceability, and change management of security requirements. These activities are crucial for mitigating vulnerabilities early in the software development life cycle, thereby supporting the development of secure systems from their inception.

Among the primary studies analyzed in this review, there is a noticeable emphasis on the use of LLMs for the classification of requirements. While such studies offer valuable insights, they often rely on similar methodologies and report comparable results, suggesting a possible saturation of this research line. Consequently, there is a compelling opportunity to advance the state of the art by exploring the use of LLMs for more complex and critical tasks, such as the automated extraction of requirements from natural language sources, the precise formulation of security requirements, and their semi-automated validation against threat scenarios. However, considerable effort is required to consolidate accurate and open databases to make this kind of research feasible.

Given the increasing adoption of generative LLMs and prompting engineering techniques, it is crucial to establish standardized metrics and robust benchmarking frameworks to enable fair and unbiased comparisons across different approaches. The lack of such standards hinders the reproducibility of studies and poses challenges to consolidating the knowledge base in this domain. Moreover, empirical evidence involving industry practitioners remains scarce, limiting the assessment of the practical applicability and

effectiveness of proposed solutions. Future research should prioritize studies that integrate experimental approaches with industry validations, fostering the transfer of knowledge from academia to real-world settings.

To ensure validity, this review followed established rapid review guidelines [10], with study selection and data extraction conducted by at least two researchers, including a third for resolving disagreements. As the field is still emerging and rapidly evolving, the conclusions reflect the current landscape and may require future updates as new evidence emerges.

ARTIFACT AVAILABILITY

Artifacts related to data collection and analysis used in this study are available in [13].

ACKNOWLEDGMENTS

The authors thank the São Paulo Research Foundation (FAPESP) for its support through processes 2024/13482-8 and 2024/19047-1, and the "Pró-Reitoria de Pesquisa e Inovação, Universidade de São Paulo" (PRPI-USP) (Grant number: 22.1.09345.01.2).

REFERENCES

[1] Abdel-Jaouad Aberkane, S. V. Broucke, G. Poels, and Georgios Georgiadis. 2024. Leveraging ChatGPT for GDPR Compliance in Requirements Engineering: A Pilot Study. *International Conference on Security, Privacy, and Anonymity in Computation, Communication, and Storage* (2024). https://doi.org/10.1109/spaccs63173.2024.00012

[2] Khubaib Amjad Alam, Hira Asif, Irum Inayat, and Saif Ur Rehman Khan. 2024. Automated Quality Concerns Extraction from User Stories and Acceptance Criteria for Early Architectural Decisions. *European Conference on Software Architecture* (2024). https://doi.org/10.1007/978-3-031-70797-1_24

[3] Abdulrahim Alhaizaey and Majed Al-Mashari. 2025. Automated Classification and Identification of Non-Functional Requirements in Agile-Based Requirements Using Pre-Trained Language Models. *IEEE Access* (2025). https://doi.org/10.1109/access.2025.3570359

[4] Waad Alhoshan, Alessio Ferrari, and Liping Zhao. 2023. Zero-shot learning for requirements classification: An exploratory study. *null* (2023). https://doi.org/10.1016/j.infsof.2023.107202

[5] Jomar Thomas Almonte, Santhosh Anitha Boominathan, and Nathalia Nascimento. 2025. Automated Non-Functional Requirements Generation in Software Engineering with Large Language Models: A Comparative Study. *arXiv.org* (2025). https://doi.org/10.48550/arxiv.2503.15248

[6] Ayah Alqurashi and Luay Alawneh. 2024. Stacked Ensemble Deep Learning for the Classification of Nonfunctional Requirements. *IEEE Transactions on Reliability* (2024). https://doi.org/10.1109/tr.2024.3513834

[7] Valentina Alto. 2024. *Building LLM Powered Applications: Create Intelligent Apps and Agents with Large Language Models.* Packt Publishing.

[8] Maurice H. Ter Beek, A. Fantechi, S. Gnesi, Gabriele Lenzini, and M. Petrocchi. 2024. Can AI Help with the Formalization of Railway Cybersecurity Requirements? *Leveraging Applications of Formal Methods* (2024). https://doi.org/10.1007/978-3-031-73709-1_12

[9] Louis-François Bouchard and Louie Peters. 2024. *Building LLMs for Production: Enhancing LLM Abilities and Reliability with Prompting, Fine-Tuning, and RAG.* Publisher not specified.

[10] B. Cartaxo, G. Pinto, and S. Soares. 2020. Rapid Reviews in Software Engineering. In *Contemporary Empirical Methods in Software Engineering*, M. Felderer and G. Travassos (Eds.). Springer, Cham. https://doi.org/10.1007/978-3-030-32489-6_13

[11] E. Casalicchio and Alberto Cotumaccio. 2024. AI-CRAS: AI-driven Cloud Service Requirement Analysis and Specification. *2024 IEEE International Conference on Cloud Engineering (IC2E)* (2024). https://doi.org/10.1109/ic2e61754.2024.00009

[12] Fabiano Dalpiaz. 2018. *Requirements data sets (user stories).* https://doi.org/10.17632/7zbk8zsd8y.1

[13] Amália Vitória de Melo, Lucas Almeida, and Lina Garcés. 2025. *Investigating the use of Large Language Models in software security requirements: Results of a literature review.* https://doi.org/10.5281/zenodo.16790764

[14] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M. Zhang. 2023. Large Language Models for Software Engineering: Survey and Open Problems . In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE).* IEEE Computer Society, Los Alamitos, CA, USA, 31–53. https://doi.org/10.1109/ICSE-FoSE59343.2023.00008

[15] Alessio Ferrari and Paola Spoletini. 2025. Formal requirements engineering and large language models: A two-way roadmap. *Information and Software Technology* 181 (2025), 107697. https://doi.org/10.1016/j.infsof.2025.107697

[16] Jameleddine Hassine. 2024. An LLM-based Approach to Recover Traceability Links between Security Requirements and Goal Models. *International Conference on Evaluation Assessment in Software Engineering* (2024). https://doi.org/10.1145/3661167.3661261

[17] Guntur Budi Herwanto, G. Quirchmayr, A. Tjoa, and Guntur Budi Herwanto. 2024. Leveraging NLP Techniques for Privacy Requirements Engineering in User Stories. *IEEE Access* (2024). https://doi.org/10.1109/access.2024.3364533

[18] Tobias Hey, Tobias Hey, Jan Keim, Jan Keim, Anne Koziolek, Anne Koziolek, Anne Koziolek, Walter F. Tichy, and Walter F. Tichy. 2020. NoRBERT: Transfer Learning for Requirements Classification. *IEEE International Requirements Engineering Conference* (2020). https://doi.org/10.1109/re48521.2020.00028

[19] Stefan Hirschmeier. 2024. CISO-BERT: Matching Information Security Requirements by Fine-Tuning the BERT Language Model. *Hawaii International Conference on System Sciences* (2024). https://doi.org/10.24251/hicss.2024.168

[20] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2024. Large Language Models for Software Engineering: A Systematic Literature Review. *ACM Trans. Softw. Eng. Methodol.* 33, 8, Article 220 (Dec. 2024), 79 pages. https://doi.org/10.1145/3695988

[21] ISO. 2022. *Information security, cybersecurity and privacy protection — Information security management systems — Requirements.* Technical Report. International Organization for Standardization.

[22] ISO. 2024. *Information technology — Security techniques — Privacy framework.* Technical Report. International Organization for Standardization.

[23] Paul Iusztin. 2024. *LLM Engineer's Handbook: Master the Art of Engineering Large Language Models from Concept to Production.* Packt Publishing.

[24] Rabi Jay. 2024. *Generative AI Apps with Langchain and Python: A Project-Based Approach to Building Real-World LLM Apps.* Apress.

[25] Muhammad Amin Khan, Mohammad Sohail Khan, I.B. Khan, Shafiq Ahmad, and Shamsul Huda. 2023. Non Functional Requirements Identification and Classification Using Transfer Learning Model. *IEEE Access* (2023). https://doi.org/10.1109/access.2023.3295238

[26] Hanyue Liu, Marina Bueno García, and Nikolaos Korkakakis. 2024. Exploring Multi-Label Data Augmentation for LLM Fine-Tuning and Inference in Requirements Engineering: A Study with Domain Expert Evaluation. In *2024 International Conference on Machine Learning and Applications (ICMLA).* 432–439. https://doi.org/10.1109/ICMLA61862.2024.00064

[27] Xianchang Luo, Yinxing Xue, Zhenchang Xing, and Jiamou Sun. 2022. PRCBERT: Prompt Learning for Requirement Classification using BERT-based Pretrained Language Models. *International Conference on Automated Software Engineering* (2022). https://doi.org/10.1145/3551349.3560417

[28] Luca Petrillo, Fabio Martinelli, A. Santone, and F. Mercaldo. 2025. Explainable Security Requirements Classification Through Transformer Models. *Future Internet* (2025). https://doi.org/10.3390/fi17010015

[29] Katia Romero Felizardo Scannavino, Elisa Yumi Nakagawa, Sandra Camargo Pinto Ferraz Fabbri, and Fabiano Cutigi Ferrari. 2017. Revisão Sistemática da Literatura em Engenharia de Software: teoria e prática. (2017).

[30] Shaden Shaar, Nikolay Babulkov, Giovanni Da San Martino, and Preslav Nakov. 2020. That is a Known Lie: Detecting Previously Fact-Checked Claims. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).* Association for Computational Linguistics, 3607–3618. https://doi.org/10.18653/v1/2020.acl-main.332

[31] Vasily Varenov, Vasily Varenov, Aydar Gabdrahmanov, and Aydar Gabdrahmanov. 2021. Security Requirements Classification into Groups Using NLP Transformers. *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)* (2021). https://doi.org/10.1109/rew53955.2021.9714713

[32] Prof. Riccardo COPPOLA Vittoria OCLEPPO. 2024-2025. *Enhancing Requirements Engineering with Large Language Models: From Elicitation and Classification to Traceability, Ambiguity Management and API Recommendation.* Ph. D. Dissertation. POLITECNICO DI TORINO.

[33] Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, and Qing Wang. 2024. Software Testing With Large Language Models: Survey, Landscape, and Vision. *IEEE Trans. Softw. Eng.* 50, 4 (April 2024), 911–936. https://doi.org/10.1109/TSE.2024.3368208

[34] Georgia Xanthopoulou, Miltiadis Siavvas, Ilias Kalouptsoglou, Dionysios Kehagias, and Dimitrios Tzovaras. 2024. Software Requirements Classification: From Bag-of-Words to Transformer. In *International Symposium on Distributed Computing and Artificial Intelligence.* Springer, 370–380.