Does Model Size Matter? A Comparison of Small and Large Language Models for Requirements Classification

Mohammad Amin Zadenoori[1 [0000−0003−4591−153X]], Vincenzo De Martino[2,3 [0000−0003−1485−4560]], Jacek Dąbrowski[4 [0000−0003−3392−0690]], Xavier Franch[3 [0000−0001−9733−8830]], and Alessio Ferrari[5 [0000−0002−0636−5663]]

[1] University of Padova, Italy
amin.zadenoori@unipd.it
[2] Software Engineering (SeSa) Lab, University of Salerno, Italy
vdemartino@unisa.it
[3] Universitat Politècnica de Catalunya, Spain
{vincenzo.de.martino, xavier.franch}@upc.edu
[4] Lero, the Research Ireland Centre for Software, University of Limerick, Ireland
jacek.dabrowski@lero.ie
[5] University College Dublin (UCD), Ireland
alessio.ferrari@ucd.ie

Abstract. [Context and motivation] Large language models (LLMs) show notable results in natural language processing (NLP) tasks for requirements engineering (RE). However, their use is compromised by high computational cost, data-sharing risks, and dependence on external services. In contrast, small language models (SLMs) offer a lightweight, locally deployable alternative. **[Question/problem]** It remains unclear how well SLMs perform compared to LLMs in RE tasks in terms of accuracy. **[Results]** Our preliminary study compares eight models, including three LLMs and five SLMs, on requirements-classification tasks using the PROMISE, PROMISE Reclass, and SecReq datasets. Our results show that although LLMs achieve an average F1 score of 2% higher than SLMs, this difference is not statistically significant. SLMs almost reach LLMs performance across all datasets and even outperform them in recall on the PROMISE Reclass dataset, despite being up to 300 times smaller. We also found that dataset characteristics play a more significant role in performance than model size. **[Contribution]** Our study contributes with evidence that SLMs are a valid alternative to LLMs for requirements classification, offering advantages in privacy, cost, and local deployability.

1 Introduction

Requirements classification is a critical task in requirements engineering (RE). It involves categorizing requirements into types such as functional/non-functional, or more fine-grained classes (e.g., security, performance) [17]. This task supports other RE activities, including requirements management and traceability [1].

However, as projects grow larger, the number of requirements can reach thousands, making manual classification labor-intensive and error-prone. This has motivated research on automating requirements classification using natural language processing (NLP) techniques [17,7]. Recent advances in large language models (LLMs), such as GPT and Claude, have transformed NLP, achieving state-of-the-art results in text understanding, classification, and generation [15]. Consequently, the RE community is increasingly exploring their use for RE tasks, particularly automated requirements classification [2].

Despite their power, LLMs raise concerns about privacy, security, and reproducibility. They are typically closed-source and cloud-hosted, which increases the risk of data exposure, as company requirements constitute confidential assets [9] and the cloud operates as an external service. Their proprietary nature also limits researchers' ability to adapt models for RE-specific needs and reduces their practicality in industrial settings. Open small language models (SLMs) offer a viable alternative for local execution on private machines or servers, enabling secure processing of sensitive data. Local deployment also lowers costs and enables easier customization and fine-tuning [13]. Furthermore, they offer advantages in terms of energy consumption [6]. However, despite their advantages, the performance comparison between SLMs and LLMs in RE tasks remains unexplored.

This study addresses that gap through a systematic comparison of eight language models. We include five SLMs with 7-8 billion parameters (*Qwen2-7B*, *Falcon-7B*, *Granite-3.2-8B*, *Ministral-8B*, and *Llama-3-8B*) and three LLMs (*GPT-5*, *Grok-4*, and *Claude-4*), with 1-2 trillion parameters. The models were selected based on the top-performing LLMs in the Hugging Face Open *LLM Leaderboard*[6] to ensure a fair comparison. We evaluated all models on three public datasets: *PROMISE* [3], a re-classification of PROMISE that we call *PROMISE Reclass* [4], and *SecReq* [10]. Our results show that SLMs perform comparably to LLMs, although the latter are 100 to 300 times larger in terms of parameters. Among all LLMs, *Claude-4* consistently achieves the highest F1 scores, including 0.81 on PROMISE, 0.80 on PROMISE Reclass, and 0.89 on SecReq. The best-performing SLM, *Llama-3-8B*, reached 0.76, 0.78, and 0.88 on the same tasks, respectively, yielding an average F1 margin of 0.02 in favor of the LLMs. Our results suggest that model size has a limited effect on classification accuracy, while SLMs offer a promising balance between performance, privacy, and resource efficiency. Furthermore, we show that performance highly depends on the dataset used for classification. **Data Availability Statement.** To promote transparency and reproducibility, we make our replication package publicly available [14].

2 Study Design

Our study aims to address the following research question **(RQ):**

> **RQ:** *What is the performance difference between SLMs and LLMs in requirements classification tasks?*

[6] `https://huggingface.co/spaces/open-llm-leaderboard`

To address our RQ, we designed a reproducible pipeline, described in Table 1. It should be noted that, although some open-source LLMs now approach or exceed 100B parameters (e.g., Falcon-180B, Mixtral-8x22B) and can be executed on personal servers, *LLMs* in this study refer exclusively to proprietary models maintained by private companies and accessed via commercial APIs (e.g., GPT-5, Claude-4, Grok-4). In contrast, *SLMs* denote openly accessible models (e.g., Qwen2-7B, Falcon-7B) that support local deployment. The evaluated SLMs range from 7B to 8B parameters, whereas LLMs can reach trillions, although their precise sizes remain undisclosed.

Table 1. Study Design Overview Following the PRIMES Framework [5]

Component	Description
Datasets	**PROMISE** [3]: Binary classification of Functional Requirements (FR) vs. Non-Functional Requirements (NFR). Composed of 625 requirements (255 FR, 370 NFR). **PROMISE Reclass** [4]: Two binary subtasks: FR vs. NFR and QR vs. Non-QR. Each requirement has two labels. Composed of 625 requirements (310 FR, 230 FR only; 382 QR, 302 QR only). **SecReq** [10]:Binary classification of Security-related (Sec) vs. Non-Security (NSec) requirements. Composed of 510 requirements (187 Sec, 323 NSec).
Models	**SLMs:** Qwen2-7B-Instruct, Falcon-7B-Instruct, Granite-3.2-8B-Instruct, Ministral-8B-Instruct-2410, Meta-Llama-3-8B-Instruct. **LLMs:** GPT-5, xAI Grok-4, Claude-4.
Prompting Strategy	**Chain-of-Thought** (CoT) plus **Few-Shot** prompting with four examples per class, which—based on our prior work [16]—emerged as the most effective among all evaluated prompting strategies. Definitions of categories are based on expert-defined RE definitions [11,8].
Parameters	Default parameters; temperature fixed at 0 for deterministic outputs.
Hardware and software	SLMs executed on Linux 6.14 server with Intel i9-13900K CPU, 128 GB RAM, NVIDIA RTX 4090 GPU, Python 3.12. LLMs accessed via commercial APIs.
Runs	Each model evaluated on all datasets, with three executions per task. Majority voting (2/3) is used for the final label. Metrics computed via macro-averaging.
Statistical test	Scheirer-Ray-Hare test, a non-parametric equivalent of two-way ANOVA.

3 Preliminary Results and Discussion

This section presents the preliminary results of our investigation, analyzing the performance of the evaluated models across the selected datasets. Table 2 reports classification performance per dataset, including average metrics, considering precision (P), recall (R), and F1. Our evaluation employs these standard metrics without assuming any specific trade-off between P and R, as their relative importance varies across application contexts.

Overall Performance. Across the evaluated classification tasks, LLMs demonstrate a consistent, albeit subtle, performance advantage over SLMs. However, a detailed examination reveals a more complex picture where SLMs are highly competitive and even excel in specific areas. While LLMs lead in average F1 score in all datasets, the performance gap is not uniform. In fact, in all datasets and metrics, we observe that there is always at least one SLM behaving better than one LLM. In the PROMISE dataset, the leading SLMs (Qwen2-7B, Falcon3-7B, and Granite-3.2) achieve an F1 score of 0.83, effectively tying with the best

Table 2. Results across all datasets using CoT ∪ Few-shot.

	PROMISE			PROMISE Reclass			SecReq		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Qwen2-7B	0.86	0.80	0.83	0.55	**0.96**	0.70	0.83	0.90	0.86
Falcon3-7B	0.86	0.80	0.83	0.58	**0.96**	0.72	0.79	0.90	0.84
Granite-3.2	0.84	0.82	0.83	0.62	0.87	0.72	0.84	0.90	0.87
Ministral-8B	0.84	0.77	0.80	0.64	0.89	0.75	0.83	0.88	0.85
Llama-3-8B	0.84	0.77	0.80	0.70	0.87	0.78	0.86	0.91	0.88
SLMs (Mean)	0.85	0.79	0.82	0.62	0.91	0.73	0.83	0.90	0.86
Grok-4	**0.88**	**0.83**	**0.85**	0.60	0.82	0.69	0.84	0.89	0.86
GPT-5o	0.85	0.81	0.83	0.68	0.88	0.77	0.85	0.90	0.87
Claude-4	0.85	0.80	0.82	**0.72**	0.90	**0.80**	**0.87**	**0.92**	**0.89**
LLMs (Mean)	0.86	0.81	0.83	0.67	0.87	0.75	0.85	0.90	0.88

LLM, Grok-4 (0.85). More notably, in the PROMISE Reclass dataset, SLMs demonstrate a remarkable advantage in Recall, with Qwen2-7B and Falcon3-7B reaching 0.96—significantly higher than any LLM. This indicates that SLMs are exceptionally effective at identifying all relevant instances within this specific context, minimizing false negatives. Furthermore, the SLM Ministral-8B achieves a Precision of 0.64 in PROMISE Reclass, outperforming the LLM Grok-4 (0.60), suggesting greater reliability in its positive predictions for that dataset.

Even in the SecReq dataset, the performance difference is minimal, with the top SLM, Llama-3-8B (F1: 0.88), closely trailing the top LLM, Claude-4 (F1: 0.89). These nuanced results indicate that while LLMs possess a slight edge in balanced overall performance (F1), SLMs are not merely competitive; they can peak higher than LLMs in isolated but crucial metrics like Recall and Precision. This suggests that SLMs have specialized strengths that make them highly capable for specific classification scenarios where minimizing certain types of errors is paramount. To summarize our performance findings in each dataset:

- **PROMISE:** Grok-4 (LLM) shows best behaviour for all metrics, but interestingly three SLMs behave similarly than the other two LLMs. As result, averages are quite close among both types.
- **PROMISE Reclass:** We notice remarkable variations of different sign in P and R, and consequently a drop in F1. LLMs behave better in P and F1 (with Claude-4 as best) but SLMs exhibit higher R.
- **SecReq:** Claude-4 also dominates in this dataset, remarkable in all metrics. Behaviour of all other LLMs and SLMs is quite close to each other, with an overall minor dominance on the side of LLMs.

Tests and Hypothesis. We next formulate and test our hypothesis. Although our RQ focuses on performance differences, here we also consider the *dataset* variable, as this could be a factor influencing the results. The null hypotheses are: (i) H_{0A} model type has no effect on performance, (ii) H_{0B} dataset has no effect on performance, and (iii) H_{0C} there is no interaction between model type and dataset. Since the sample distributions did not meet the assumptions of normality, as assessed by the **Shapiro-Wilk test**, we used the **Scheirer-Ray-Hare test**, a non-parametric equivalent of two-way ANOVA (see results in Table 3). The post-hoc tests are follow-up analyses that break down the overall effects reported in 3, specifying whether differences between model types or datasets are

Table 3. Scheirer-Ray-Hare Test Results for Model Performance (F1 Score). Effect sizes: $\eta^2_H = 0.01$ (small), 0.06 (medium), 0.14 (large).

Hp	Variable	Effect Size (η^2_H)	p-value	Reject/Fail to Reject
H_{0A}	Model Type	0.04	0.296	Fail to Reject H_{0A}
H_{0B}	Dataset	0.63	**<0.001**	**Reject** H_{0B}
H_{0C}	Model Type × Dataset	0.001	0.790	Fail to Reject H_{0C}

statistically significant within individual comparisons. Post-hoc analyses were conducted using pairwise Mann-Whitney U tests with Bonferroni correction.

Post-hoc Model Type Comparisons within each Dataset:

- PROMISE: U = 7.0, p = 0.134 (Not Significant)
- PROMISE_Reclass: U = 11.0, p = 0.764 (Not Significant)
- SecReq: U = 7.0, p = 0.291 (Not Significant)

Post-hoc Pairwise Dataset Comparisons (Bonferroni-corrected):

- PROMISE_Reclass vs. PROMISE: U = 6.50, p = 0.0084 (Significant)
- PROMISE_Reclass vs. SecReq: U = 0.00, p = 0.0009 (Significant)
- PROMISE vs. SecReq: U = 3.50, p = 0.0031 (Significant)

Interpretation of Findings. Based on the results, the following conclusions can be derived.

1. **Model Type Effect (SLM vs. LLM):** The analysis confirms no statistically significant main effect of model type on F1 score (H = 1.09, p = 0.296, and small effect size $\eta^2_H = 0.04$). Despite descriptive statistics showing slightly higher mean F1 scores for LLMs (0.818) compared to SLMs (0.793), this difference is not large enough to be considered statistically significant.
2. **Dataset Effect:** We found a highly significant main effect of the dataset on F1 score (H = 17.53, p < 0.001, and large effect size $\eta^2_H = 0.63$). Post-hoc tests revealed that all pairwise differences between datasets are statistically significant, with a clear performance hierarchy: models performed worst on **PROMISE_Reclass** (Median F1 = 0.730), better on **PROMISE** (Median F1 = 0.805), and best on **SecReq** (Median F1 = 0.865).
3. **Interaction Effect (Model Type × Dataset):** Critically, the **interaction between Model Type and Dataset was not significant** (H = 0.47, p = 0.790, and small effect size $\eta^2_H = 0.001$). Since the interaction is not significant, the effect of Dataset observed above is consistent across models—**all models are affected by datasets in roughly the same way**. This is further supported by the post-hoc tests within each dataset, which all failed to show a significant difference between SLMs and LLMs on any individual dataset.

It should be noted that, while the analysis found no statistically significant performance difference between SLMs and LLMs ($p = 0.296$), the consistent directional advantage for LLM type (represented by the mean of all models) across all datasets might indicate a high probability of Type II error (i.e., we fail to reject H_{0A}, although there is a difference between the performance of SLMs and LLMs). The limited sample size of eight models substantially reduces

statistical power, likely obscuring genuine performance differences. Nevertheless, even if the difference is confirmed to be significant by future studies, we argue that a loss of 2% can be considered acceptable, given the advantages of SLMs in data privacy and resource efficiency.

> 👉 **Answer to RQ:** LLMs show a slight edge in performance compared to SLMs, leading by 2% in F1. The difference is, however, not statistically significant. Furthermore, when considering the recall metric on specific datasets, SLMs tend to outperform LLMs.

Threats to Validity. *Construct.* We have used traditional metrics for evaluation, and not accounted for F_β. Further experiments are required to identify the appropriate β value for our context. These were not conducted, given the preliminary nature of the study. *Internal.* The same prompt was applied across all models to ensure consistency; however, relying on a single prompt, even if informed by prior studies [16], may affect internal validity since model performance could depend on prompt phrasing [2]. Internal validity is also affected by the variability of model output. To address this, we repeated each run three times, and performed majority voting. There is also a risk of data leakage, as some datasets may have been included in the pre-training corpora of the evaluated LLMs, potentially inflating their performance. This could not be entirely mitigated. *External.* We have considered the binary classification task, so our results might not generalise to other RE tasks. On the other hand, we have used three different datasets, which increase the generalisability of our results, at least for the task at hand *Conclusion.* We performed appropriate statistical tests to check our conclusions. However, the limited sample size might have led to Type II errors. This is justified by the preliminary nature of the experiments. To prevent multiple comparison problems arising from different dependent variables, we used only the F1 metric for statistical analysis, which might not account for all performance nuances. To address this, we have complemented the results with observations on precision and recall for specific cases, and reported the values in Table 2.

4 Conclusion and Roadmap

Our study compared LLMs and SLMs performance in requirements classification, showing that the formed gives only marginal F1 improvement (2%) over the latter. This suggests that, at least for this RE tasks, companies can profit for the advantages of SLMs in terms of data privacy and resource demands. The following research avenues will be explored in our future research.

Explainability. While LLMs and SLMs have shown strong and comparable performance in requirements classification, such tasks do not fully leverage their generative capabilities: classification alone captures the outcome of reasoning but not the reasoning process itself. To exploit the full potential of these models, we should move beyond simple label prediction and focus on generating *explanations* that justify classifications. Explanations are particularly important in RE, as they enable practitioners to make informed decisions based on the rationale behind

classifications rather than on predictions alone. Understanding why a requirement is categorized in a certain way supports confidence in automated analyses, as well as more informed downstream tasks, e.g., traceability, and auditing.

Different RE Tasks and Hybrid Pipelines. We plan to extend the evaluation to additional RE tasks such as traceability, model generation, and requirements elicitation, which have already seen interest in LLM for RE [15]. Special attention will be given to tasks that require the generation of artifacts, as these are the ones where generative models can offer a true competitive advantage with respect to more traditional machine learning (ML) solutions. Experimenting with SLMs and LLMs in different tasks will potentially enable the identification of hybrid pipelines for RE workflows, where each model type is used for the task or phase where it brings the best advantage, e.g., traditional ML models for classification, SLMs for explanation, and LLMs for supporting analyst interaction and iterative refinement through conversational assistance.

Energy Consumption. Sustainability of AI computation is a major concern, as LLMs can have substantial environmental impact—although still lower than manual labour [12]. Further research is needed to quantify the energy footprint of RE-specific tasks, comparing SLMs and LLMs. This includes evaluating trade-offs between model performance and energy efficiency, especially in local vs. cloud-based deployments of SLMs and LLMs [6].

Execution Speed. Systematically studying speed differences between SLMs and LLMs is also an important avenue to consider, as RE activities need to be accurate but also fast, to cope with the ever-changing nature of stakeholder demands. In our study, Claude-4 is the fastest on PROMISE Reclass (300s), Grok-4 leads on PROMISE (162s), and GPT-5o is the quickest on SecReq (138s), while the computation time of SLMs is on average 400s. This gap is expected, given that proprietary LLMs are deployed on high-performance cloud infrastructure, whereas the SLMs in this study were executed on a single local server given in Table 1. Although the execution speed is specific to our set-up and can widely vary, we believe that these numbers give an indication of the order of magnitude of the time difference that a small-medium company——which can be considered similar in terms of resources to a research lab such as ours——can expect when using local SLMs. Further studies are needed to understand what speed range is practically acceptable for practitioners and how to optimise execution efficiency.

Tool Support. Deployment of LLMs in companies is a major pain point, and comparison between SLM/LLM solutions can be cumbersome. As future work, we envision the development of a tool that enables practitioners to flexibly select between locally hosted SLMs and API-based LLMs for requirements classification. The tool will adapt model selection and prompt strategies according to specific constraints, such as data privacy, computational resources, and task complexity, thus supporting context-aware and efficient RE workflows.

Acknowledgements This work was supported in part by project EOSS6-0000000644 from the Chan Zuckerberg Initiative. Grok-4 (https://x.ai/grok) was used to improve the readability, and the authors remain fully responsible for the final content.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alhoshan, W., Ferrari, A., Zhao, L.: Zero-shot learning for requirements classification: An exploratory study. Information and Software Technology **159**, 107202 (2023)
2. Alhoshan, W., Ferrari, A., Zhao, L.: How effective are generative large language models in performing requirements classification? (2025), https://arxiv.org/abs/2504.16768
3. Cleland-Huang, J., Settimi, R., Zou, X., Solc, P.: Automated classification of non-functional requirements. Requirements Engineering **12**(2), 103–120 (Apr 2007). https://doi.org/10.1007/s00766-007-0045-1
4. Dalpiaz, F., Dell'Anna, D., Aydemir, F., Cevikol, S.: Requirements classification with interpretable machine learning and dependency parsing. pp. 142–152 (09 2019). https://doi.org/10.1109/RE.2019.00025
5. De Martino, V., Castaño, J., Palomba, F., Franch, X., Martínez-Fernández, S.: A framework for using llms for repository mining studies in empirical software engineering. In: 2025 IEEE/ACM International Workshop on Methodological Issues with Empirical Studies in Software Engineering (WSESE). pp. 6–11. IEEE (2025)
6. De Martino, V., Zadenoori, M.A., Franch, X., Ferrari, A.: Green prompt engineering: Investigating the energy impact of prompt design in software engineering (2025)
7. Dąbrowski, J., Letier, E., Perini, A., Susi, A.: Analysing app reviews for software engineering: a systematic literature review. Empirical Software Engineering **27**(2), 43 (2022). https://doi.org/10.1007/s10664-021-10065-7
8. Ernst, N.A., Mylopoulos, J.: On the perception of software quality requirements during the project lifecycle. In: Wieringa, R., Persson, A. (eds.) REFSQ. pp. 143–157. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
9. Ferrari, A., Dell'Orletta, F., Esuli, A., Gervasi, V., Gnesi, S., et al.: Natural language requirements processing: a 4D vision. IEEE SOFTWARE **34**(6), 28–35 (2017)
10. Knauss, E., Houmb, S., Schneider, K., Islam, S., Jürjens, J.: Supporting requirements engineers in recognising security issues. In: REFSQ 2011. pp. 4–18. https://doi.org/10.1007/978-3-642-19858-8_2
11. Olsson, T., Sentilles, S., Papatheocharous, E.: A systematic literature review of empirical research on quality requirements. Requirements Engineering **27**(2), 249–271 (Jun 2022). https://doi.org/10.1007/s00766-022-00373-9
12. Ren, S., Tomlinson, B., Black, R.W., Torrance, A.W.: Reconciling the contrasting narratives on the environmental impact of large language models. Scientific Reports **14**(1), 26310 (2024)
13. Wang, F., et al.: A comprehensive survey of small language models in the era of large language models: Techniques, enhancements, applications, collaboration with llms, and trustworthiness. preprint arXiv:2411.03350 (2024)
14. Zadenoori, A.: Comparison of small and large language models for requirements classification (Oct 2025), https://doi.org/10.5281/zenodo.17339105
15. Zadenoori, M.A., Dąbrowski, J., Alhoshan, W., Zhao, L., Ferrari, A.: Large language models (LLMs) for requirements engineering (RE): A systematic literature review (2025), https://arxiv.org/abs/2509.11446
16. Zadenoori, M.A., Zhao, L., Alhoshan, W., Ferrari, A.: Automatic prompt engineering: The case of requirements classification. In: Hess, A., Susi, A. (eds.) REFSQ. pp. 217–225. Springer Nature Switzerland, Cham (2025)
17. Zhao, L., Alhoshan, W., Ferrari, A., Letsholo, K.J., Ajagbe, M.A., Chioasca, E.V., Batista-Navarro, R.T.: Natural language processing for requirements engineering: A systematic mapping study. ACM Comput. Surv. **54**(3) (Apr 2021)