Vulnerabilities of Small Language Models: A Systematic Literature Review

Daniel Dias
Department of Informatics Engineering
Polytechnic School of Porto
Porto, Portugal
1240145@isep.ipp.pt

***Abstract*—Context: Small Language Models (SLMs), typically defined as models with fewer than 7 billion parameters, have emerged as efficient alternatives to Large Language Models (LLMs) for deployment in resource-constrained environments such as edge devices, mobile applications, and IoT systems. As their adoption accelerates, understanding their security vulnerabilities becomes critical. Goal: This systematic literature review aims to identify, categorize, and analyze the security vulnerabilities specific to small language models, examining how these vulnerabilities differ from those observed in larger models and evaluating proposed mitigation strategies. Method: We conducted a systematic literature review following PRISMA guidelines, searching multiple databases including IEEE Xplore, ACM Digital Library, arXiv, and Semantic Scholar. Studies were selected based on predefined inclusion and exclusion criteria, with quality assessment performed using established protocols. Results: Our analysis reveals that SLMs exhibit distinct vulnerability profiles compared to their larger counterparts, with particular susceptibility to prompt injection, jailbreaking, and adversarial attacks. The reduced parameter count often limits the implementation of safety mechanisms, creating unique security challenges. Conclusion: This review provides the first comprehensive taxonomy of SLM-specific vulnerabilities and identifies critical gaps in current research, offering a foundation for future security research in small language models.**

***Index Terms*—Small Language Models, Security Vulnerabilities, Prompt Injection, Jailbreaking, Adversarial Attacks, Systematic Literature Review**

I. Introduction

The field of natural language processing has undergone a fundamental transformation with the advent of transformer-based architectures [1] and the subsequent development of Large Language Models. Systems such as GPT [2], Claude, and Gemini have demonstrated remarkable capabilities in reasoning, code generation, and creative tasks, driven by scaling to hundreds of billions of parameters [3], [4]. These foundation models have reshaped expectations for artificial intelligence and attracted substantial industrial investment [5].

Yet a critical countertrend is emerging: the rise of small language models designed for practical deployment and mitigating private data leakage. Small language models, typically defined as transformer-based architectures with around 7-10 billion parameters [6], [7], have emerged as compelling alternatives that address fundamental limitations of their larger counterparts. SLMs require substantially less memory and processing power, enabling deployment on hardware ranging from consumer GPUs to mobile devices and IoT systems, with proportionally reduced operational costs [6], [8]. Edge deployment enables real-time inference without network latency, critical for autonomous systems, industrial automation, and responsive interfaces. Privacy-sensitive applications benefit from local inference, eliminating the need to transmit confidential data to cloud services. Models such as Microsoft's Phi-2 [9], TinyLlama [10], and Google's Gemma [11] have demonstrated that compact architectures can achieve remarkable performance on targeted tasks, challenging assumptions about the necessity of scale. Recent work from NVIDIA researchers argues that small language models represent the future of agentic AI systems, as the specialized and repetitive tasks characteristic of such systems are better suited to smaller, more efficient models [7]. This perspective reflects a broader recognition that practical considerations of cost, latency, and deployment constraints increasingly favor compact alternatives.

However, the rapid adoption of small language models has outpaced systematic investigation of their security properties, creating significant risks for deploying organizations. Zhang et al. found that nearly half of the 63 SLMs they tested exhibited attack success rates exceeding 40% against jailbreak attempts, while over one-third failed to resist even straightforward harmful requests [12]. The fundamental challenge lies in the tension between model capacity and security implementation: safety mechanisms that function effectively in models with hundreds of billions of parameters may degrade or fail entirely when compressed into smaller architectures [13]. Yi et al. documented how compression techniques commonly used to create deployable SLMs can compromise security robustness, revealing "submerged threats" that emerge from efficiency optimizations [14]. Edge deployment introduces additional attack surfaces, as models operating in physically accessible environments face threats ranging from adversarial input injection [15] to direct model extraction [16], [17]. The combination of reduced defensive capacity and expanded attack surface creates a security landscape that demands dedicated investigation, distinct from the extensive but LLM-focused security literature [18].

II. Methodology

This systematic literature review follows the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 guidelines [19] and adopts established protocols for conducting systematic reviews in software engineering [20]. The methodology encompasses research question formulation, systematic search strategy development, study selection through predefined criteria, and thematic data synthesis.

A. Research Questions

To address the main research question—*What is the current state of knowledge regarding security vulnerabilities in Small Language Models, and how do these vulnerabilities manifest differently compared to Large Language Models?*—the most relevant and recent literature will be analyzed according to four sub-questions (see Table I).

TABLE I
Research Questions

RQ	Description
RQ1	What types of security vulnerabilities have been identified in Small Language Models, and how are they characterized and categorized in the existing literature?
RQ2	How do vulnerabilities in SLMs differ from those in LLMs in terms of attack success rates, exploitability, and severity?
RQ3	What specific architectural or deployment characteristics of SLMs contribute to their unique vulnerability profiles?
RQ4	What mitigation strategies and defense mechanisms have been proposed for SLM vulnerabilities, and what empirical evidence supports their effectiveness?

The **first sub-question** focuses on identifying the types of security vulnerabilities that have been documented in small language models. This analysis will provide insight into current attack taxonomies, including jailbreaking, prompt injection, backdoor attacks, adversarial examples, membership inference, and model extraction attacks targeting SLMs.

The **second sub-question** aims to explore the comparative security landscape between small and large language models. This is crucial for understanding whether the reduced parameter count of SLMs correlates with increased susceptibility to attacks, and how attack success rates vary across model scales.

The **third sub-question** investigates the root causes of SLM-specific vulnerabilities, examining how architectural decisions such as compression, quantization, and knowledge distillation affect security properties. Additionally, this includes evaluating how edge deployment constraints and resource limitations impact the implementation of safety mechanisms.

Finally, the **fourth sub-question** focuses on identifying the defense mechanisms and mitigation strategies proposed in the literature for protecting SLMs. This includes evaluating both theoretical proposals and empirically validated countermeasures, assessing their effectiveness across different vulnerability categories.

B. Search Strategy

The search strategy employed in this review was designed to identify the most relevant and up-to-date studies available in the scientific literature related to the research topic. The process was structured into three main stages:

1) **Definition of search sources:** selection of recognized academic databases and scientific repositories to ensure comprehensive coverage of relevant publications.
2) **Definition of search terms:** inclusion of keywords and logical combinations that accurately reflect the core concepts of the study, such as "small language model", "vulnerability", "jailbreak", and "adversarial attack".
3) **Study selection and data extraction:** application of inclusion and exclusion criteria to guarantee the relevance, quality, and methodological rigor of the analyzed works.

This structured approach ensures that the review process remains transparent, replicable, and focused on collecting the most significant contributions to the research question.

1) Definition of Search Sources: The first step of the search strategy was to identify and define which sources would be considered while conducting the SLR. For this study, searches were carried out in several major electronic databases (see Table II). These databases were selected due to their broad coverage of peer-reviewed journal articles, conference proceedings, and preprints in the fields of computer science, artificial intelligence, natural language processing, and cybersecurity, which are directly relevant to the topic of small language model vulnerabilities.

TABLE II
Electronic Databases

ID	Database	URL
ED1	arXiv	https://arxiv.org/
ED2	Semantic Scholar	https://semanticscholar.org/
ED3	ACL Anthology	https://aclanthology.org/
ED4	IEEE Xplore	https://ieeexplore.ieee.org/
ED5	ACM Digital Library	https://dl.acm.org/

Additionally, gray literature sources including technical reports from NVIDIA Research, Microsoft Security, and Anthropic were consulted to capture emerging findings not yet published in peer-reviewed venues.

2) Definition of Search Terms: The second step of the search strategy involved defining a set of search strings that accurately reflected the research questions formulated for this study. The terms were derived from the main concepts present in the research scope: small language models, security vulnerabilities, and attack/defense mechanisms. Boolean operators and synonyms were used to broaden the search and ensure coverage across different scientific databases.

Table III presents the search terms used to identify studies related to small language models, including both generic descriptors and specific model architectures.

TABLE III
SEARCH TERMS: MODEL IDENTIFICATION

Category	Search Terms
Generic SLM Terms	"small language model" OR "SLM" OR "lightweight language model" OR "edge language model" OR "compact language model" OR "efficient language model"
Decoder-only Models	"Phi-2" OR "Phi-3" OR "Gemma" OR "TinyLlama" OR "Llama 7B" OR "Mistral 7B" OR "Qwen"
Encoder-only Models	"MobileBERT" OR "DistilBERT" OR "MiniLM" OR "ALBERT" OR "TinyBERT"

Table IV presents the search terms used to capture security vulnerabilities and attack types relevant to language models.

TABLE IV
SEARCH TERMS: SECURITY AND VULNERABILITIES

Category	Search Terms
General Security	"vulnerability" OR "attack" OR "security" OR "robustness" OR "adversarial"
Prompt-based Attacks	"jailbreak" OR "prompt injection" OR "prompt leaking" OR "guardrail bypass"
Training-time Attacks	"backdoor" OR "poisoning" OR "data poisoning" OR "trojan"
Privacy Attacks	"membership inference" OR "data extraction" OR "model stealing" OR "model extraction"

The search strings were defined to cover four main pillars of the research scope:

- **Generic SLM Terms:** to capture studies on compact transformer architectures using general terminology such as "small language model" or "lightweight language model".
- **Specific Model Architectures:** focused on identifying studies related to specific models commonly deployed in resource-constrained environments, distinguishing between encoder-only architectures (BERT variants) and decoder-only architectures (Phi, Gemma, TinyLlama, Llama 7B).
- **General Security and Prompt-based Attacks:** intended to include studies addressing security weaknesses and inference-time attack vectors, such as jailbreaking and prompt injection.
- **Training-time and Privacy Attacks:** to capture studies on attacks that occur during model training (backdoors, poisoning) or that compromise data privacy (membership inference, model extraction).

The final search query combined these categories using Boolean operators as follows:

(Generic SLM Terms **OR** Decoder-only Models **OR** Encoder-only Models) **AND** (General Security **OR** Prompt-based Attacks **OR** Training-time Attacks **OR** Privacy Attacks)

Date filtering was restricted to publications from January 2020 to December 2025.

C. Inclusion and Exclusion Criteria

Studies were selected based on the following predefined criteria, which were established prior to conducting the search to ensure objectivity and reproducibility.

Inclusion Criteria:

- Published between January 2020 and December 2025
- Focuses on Small Language Models defined as models with fewer than 10 billion parameters, following the NVIDIA definition [7]
- Addresses security vulnerabilities, attacks, adversarial robustness, or defense mechanisms
- Peer-reviewed publication or high-quality preprint with substantial methodology
- Published in English

Exclusion Criteria:

- Focuses exclusively on large models ($>$10B parameters) without demonstrated applicability to SLMs
- Contains no security or vulnerability component
- Non-English publication
- Duplicate publication of the same study
- Full text not accessible
- Workshop papers or extended abstracts without sufficient methodological detail

D. Study Selection Process

The study selection followed a three-stage screening process consistent with PRISMA guidelines. In the first stage, titles of all retrieved records were screened against inclusion criteria, removing obviously irrelevant studies. The second stage involved abstract screening, where remaining studies were evaluated based on their abstracts to assess relevance to SLM security. The third stage comprised full-text assessment, where complete papers were reviewed to confirm eligibility and extract detailed information.

E. Data Extraction and Synthesis

Data extraction was performed using a standardized form capturing: study metadata (authors, year, venue, DOI), model characteristics (architecture, parameter count, quantization level), vulnerability type (jailbreak, prompt injection, backdoor, adversarial, membership inference, model extraction), attack methodology and success rates, defense mechanisms evaluated, and key findings. The extracted data were synthesized thematically, organizing findings by vulnerability categories rather than individual studies, enabling cross-study comparison and identification of consensus findings and research gaps.

F. PRISMA Flow

Figure 1 illustrates the study selection process following PRISMA 2020 guidelines. The initial database search yielded 72 records distributed across arXiv (n=26), Semantic Scholar (n=43), and ACL Anthology (n=3), with 3 additional records identified through gray literature sources (NVIDIA Research, Microsoft Security, Anthropic), totaling 75 records. After removing 7 duplicates identified through DOI and title matching, 68 unique records underwent title and abstract screening. Of these, 3 were excluded: 2 for publication date outside the specified range (pre-2020), and 1 for out-of-scope focus on using LLMs for security testing rather than vulnerabilities in SLMs. The remaining 65 studies proceeded to full-text eligibility assessment and were included in the final synthesis.

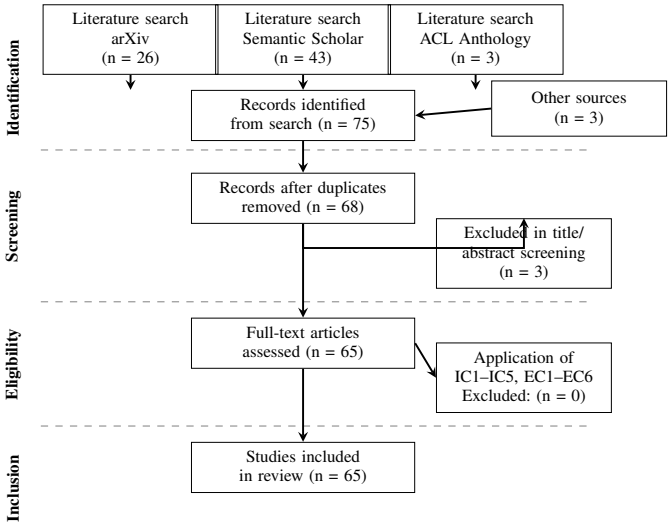

Fig. 1. PRISMA 2020 flow diagram showing study selection process.

III. Results

This section will present the findings of the systematic review organized by thematic categories.

A. Study Selection

B. Vulnerability Categories

C. Mitigation Strategies

IV. Discussion

This section will interpret the findings and discuss their implications for research and practice.

A. Key Findings

B. Implications for Practice

C. Limitations

V. Conclusions

This section will summarize the key contributions and outline future research directions.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[2] J. Achiam *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2024.

[3] T. Brown *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[4] W. X. Zhao *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[5] R. Bommasani *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[6] Z. Lu, X. Li, D. Cai, R. Yi, F. Liu, X. Zhang, N. D. Lane, and M. Xu, "Small language models: Survey, measurements, and insights," *arXiv preprint arXiv:2409.15790*, 2024.

[7] P. Belcak, G. Heinrich, S. Diao, Y. Fu, X. Dong, S. Muralidharan, Y. C. Lin, and P. Molchanov, "Small language models are the future of agentic ai," *arXiv preprint arXiv:2506.02153*, 2025.

[8] J. Tang *et al.*, "Demystifying small language models for edge deployment," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. [Online]. Available: https://aclanthology.org/2025.acl-long.718/

[9] M. Javaheripi *et al.*, "Phi-2: The surprising power of small language models," *Microsoft Research Blog*, 2023, available at: https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/.

[10] P. Zhang, G. Zeng, T. Wang, and W. Lu, "Tinyllama: An open-source small language model," *arXiv preprint arXiv:2401.02385*, 2024.

[11] Gemma Team, "Gemma: Open models based on gemini research and technology," *arXiv preprint arXiv:2403.08295*, 2024.

[12] W. Zhang, H. Xu, Z. Wang, Z. He, Z. Zhu, and K. Ren, "Can small language models reliably resist jailbreak attacks? a comprehensive evaluation," *arXiv preprint arXiv:2503.06519*, 2025.

[13] Z. Li *et al.*, "On jailbreaking quantized language models through fault injection attacks," *arXiv preprint arXiv:2507.03236*, 2025.

[14] S. Yi, T. Cong, X. He, Q. Li, and J. Song, "Beyond the tip of efficiency: Uncovering the submerged threats of jailbreak attacks in small language models," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, arXiv preprint arXiv:2502.19883.

[15] Y. Wang *et al.*, "A survey of adversarial defences and robustness in nlp," *arXiv preprint arXiv:2203.06414*, 2022.

[16] N. Carlini, D. Paleka, K. D. Dvijotham, T. Steinke, J. Hayase, A. F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conber, E. Wallace, D. Rolnick, and F. Tramèr, "Stealing part of a production language model," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, iCML 2024 Best Paper.

[17] Y. Yao *et al.*, "A survey on model extraction attacks and defenses for large language models," *arXiv preprint arXiv:2506.22521*, 2025, to appear in KDD 2025.

[18] S. Xu, X. Zhou, and Z. Hu, "Jailbreak attacks and defenses against large language models: A survey," *arXiv preprint arXiv:2407.04295*, 2024.

[19] M. J. Page *et al.*, "The prisma 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, p. n71, 2021.

[20] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *Technical Report EBSE-2007-01, Keele University and Durham University*, 2007.