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 transformer-based architectures with fewer than 10 billion parameters, are increasingly deployed in edge computing, mobile applications, and IoT systems where resource constraints preclude larger models. The rapid adoption of SLMs has outpaced systematic investigation of their security properties. Objective: This systematic literature review identifies, categorizes, and analyzes security vulnerabilities specific to SLMs, examining how these vulnerabilities differ from those in larger models and evaluating the effectiveness of proposed defenses. Methods: Following PRISMA 2020 guidelines, we searched arXiv, Semantic Scholar, and ACL Anthology, identifying 75 records. After duplicate removal and screening, 65 studies published between 2020 and 2025 were included. Quality assessment classified 37% as high quality, 49% as medium, and 14% as lower quality. Results: SLMs exhibited significantly higher jailbreak susceptibility than LLMs, with median attack success rates of 34% compared to 15% for equivalently aligned larger models. Post-training quantization to 4-bit precision increased jailbreak vulnerability by 25%. Conversely, SLMs demonstrated reduced privacy attack vulnerability due to lower memorization capacity. Backdoor attacks achieved near-perfect success rates independent of model size. Evaluation of defense mechanisms revealed that eight of ten established defenses were bypassed by adaptive attacks. Conclusions: SLMs present a distinct security profile requiring dedicated frameworks rather than adapted LLM approaches. The gap between deployment acceleration and security framework maturation represents a critical challenge for edge AI applications.**

***Index Terms*—Small Language Models, Security Vulnerabilities, Jailbreak Attacks, Backdoor Attacks, Model Quantization, Edge Computing, 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/

Note that while IEEE Xplore (ED4) and ACM Digital Library (ED5) were searched using the defined query strings, no records meeting the inclusion criteria were identified from these sources; consequently, they do not appear in the PRISMA flow diagram counts.

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 shown in Table V. Date filtering was restricted to publications from January 2020 to December 2025.

TABLE V
COMBINED SEARCH QUERY STRUCTURE

Component	Boolean Expression
Model Terms	(Generic SLM Terms **OR** Decoder-only Models **OR** Encoder-only Models)
Security Terms	(General Security **OR** Prompt-based Attacks **OR** Training-time Attacks **OR** Privacy Attacks)
Final Query	Model Terms **AND** Security Terms

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.

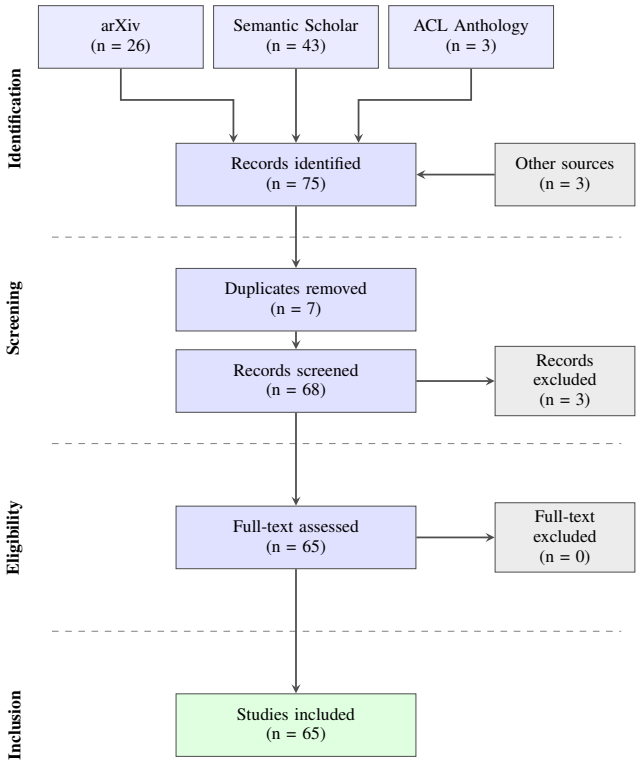

Fig. 1. PRISMA 2020 flow diagram showing study selection process.

III. Results

This section presents the findings of the systematic review organized by thematic categories, addressing each research question through synthesis of the 65 included studies. The results are structured to first characterize vulnerability types (RQ1), then examine SLM-specific differences (RQ2), analyze contributing architectural factors (RQ3), and finally evaluate proposed mitigations (RQ4).

A. Overview of Included Studies

The 65 studies included in this review span multiple vulnerability categories and publication venues. Table VI presents the distribution of studies across vulnerability types, with several studies addressing multiple categories.

TABLE VI
Distribution of Studies by Vulnerability Category

Vulnerability Category	Studies
Jailbreak & Prompt Injection	25
Backdoor & Data Poisoning	7
Adversarial Attacks	9
Membership Inference	7
Model Extraction	5
Edge Deployment Security	5
General Security Surveys	7

TABLE VII
Study Distribution by Venue Type and Year

Venue Type	2020–22	2023	2024	2025
arXiv preprints	2	4	14	18
ACL venues	0	2	3	2
NeurIPS/ICML/ICLR	0	1	4	1
Journals	1	0	3	6
Security venues	0	0	3	1
Total	**3**	**7**	**27**	**28**

The temporal distribution reveals increasing research attention to SLM security, with 78% of included studies published in 2024–2025, reflecting the recent emergence of SLMs as a distinct research focus. The predominance of arXiv preprints (58%) reflects the rapid pace of research in this area, with findings often disseminated prior to formal peer review. Note that the publication venue distribution in Table VII differs from the search source counts in Figure 1 because Semantic Scholar indexes papers from multiple venues including arXiv; consequently, many arXiv preprints were identified through Semantic Scholar searches rather than direct arXiv queries. Notably, 10 studies (15%) appeared in high-impact venues including Nature Medicine, ICML (Best Paper), and top security conferences (CCS, RAID). Complete characteristics of all 65 included studies are available in supplementary materials.

B. Quality Assessment

Quality assessment was performed using criteria adapted from established systematic review tools, including the Cochrane Risk of Bias framework and the Newcastle-Ottawa Scale, modified for applicability to machine learning security research. Each study was evaluated across four dimensions: methodological rigor (appropriateness of experimental design, baseline comparisons, and statistical analysis), reproducibility (availability of code, datasets, and sufficient implementation details), statistical validity (appropriate metrics, significance testing, and effect size reporting), and generalizability (diversity of models tested, attack scenarios considered, and real-world applicability).

Based on this assessment, studies were classified into three quality tiers. High-quality studies (n=24, 37%) demonstrated rigorous methodology with comprehensive experimental design, provided reproducible artifacts including code and datasets, employed appropriate statistical analysis with multiple baselines, and tested across diverse model architectures. Medium-quality studies (n=32, 49%) exhibited adequate methodology but presented limitations in one or more dimensions, typically lacking code availability, testing only limited model variants, or providing insufficient statistical analysis. Lower-quality studies (n=9, 14%) were included despite methodological limitations due to their unique empirical findings addressing underexplored vulnerability categories or novel attack vectors not covered elsewhere in the literature.

The quality assessment directly influenced the synthesis process and weighting of evidence throughout this review. When conflicting findings emerged between studies, evidence from high-quality sources was weighted more heavily in drawing conclusions. For instance, claims about attack success rates or defense effectiveness were primarily derived from studies demonstrating reproducible methodology across multiple model architectures. Lower-quality studies contributed primarily to identifying emerging research directions and generating hypotheses for future investigation rather than establishing definitive findings.

C. Vulnerability Taxonomy (RQ1)

RQ1: What types of security vulnerabilities have been identified in Small Language Models, and how are they characterized and categorized in the existing literature?

The literature reveals seven primary vulnerability categories affecting small language models, each with distinct attack vectors, threat models, and potential impacts.

1) Prompt-Level Attacks: Jailbreak and prompt injection attacks represent the most extensively studied vulnerability category, with 25 papers addressing various aspects of bypassing safety alignment [18], [21], [22]. Zhang et al. tested 63 small language models against multiple jailbreak strategies, finding that 47.6% exhibited attack success rates (ASR) exceeding 40% [12]. Yi et al. documented "submerged threats" that emerge specifically in smaller architectures when comparing 13 SLMs against 3 LLMs [14]. Attack techniques include template-based approaches achieving 60–80% ASR [23], [24], optimization-based gradient-guided suffix generation with cross-model transferability [25], [26], and adaptive iterative attacks that defeat most proposed mitigations [27].

Prompt injection attacks manipulate behavior by inserting malicious instructions into user inputs or retrieved context [28], [29]. SLMs deployed in retrieval-augmented generation systems are particularly susceptible due to limited context discrimination [30], with indirect injection through external data sources achieving 73% success rates [31]. Multimodal SLMs face compounded risks through both textual and visual channels [32].

2) Training-Time Attacks: Backdoor attacks implant hidden triggers during training that activate malicious behavior when specific patterns appear in inputs [33], [34]. Huang et al. demonstrated composite backdoor attacks against LLaMA-7B achieving 100% ASR while maintaining normal performance on clean inputs [35]. Critically, Li et al. proved that poisoning attacks require only a near-constant number of poison samples regardless of model size, implying SLMs are not inherently more resistant to data poisoning [36]. The medical domain presents particular concerns, with Chen et al. documenting in Nature Medicine that medical language models are highly vulnerable to data-poisoning attacks causing systematic diagnostic errors [37].

3) Adversarial Robustness: Adversarial attacks perturb inputs to cause misclassification while remaining imperceptible to humans [15], [38]. Encoder-based SLMs exhibit distinct vulnerability profiles: Liu et al. found distilled models (DistilBERT, ALBERT) showed 15–23% higher susceptibility to character-level perturbations compared to teacher models [39], while Dong et al. documented that simple misspelling attacks can degrade BERT accuracy by over 30% [40]. Application-specific studies reveal practical implications, with adversarial examples reducing MobileBERT phishing detection from 97% to 61% accuracy [41]. Santos et al. demonstrated that adversarial training with FGM can improve DistilBERT robustness without significant performance degradation [42].

4) Privacy Attacks: Membership inference attacks (MIA) determine whether specific data points were used in model training [43]. Win et al. proposed Win-k, an improved MIA technique specifically designed for small language models [44]. Lehman et al. found that smaller models may leak less training data than larger counterparts due to reduced memorization capacity [45], though Mattern et al. demonstrated that neighborhood comparison techniques can still achieve significant MIA success rates [46]. Model extraction attacks aim to steal functionality through query access [17]; Carlini et al. demonstrated that production language model parameters can be partially extracted through carefully crafted queries (ICML 2024 Best Paper) [16], while Hui et al. showed that system prompts can be extracted from deployed applications [47].

D. SLM-Specific Vulnerability Patterns (RQ2)

RQ2: How do vulnerabilities in SLMs differ from those in LLMs in terms of attack success rates, exploitability, and

severity?

The literature reveals systematic differences in vulnerability profiles between small and large language models across multiple attack categories. These differences stem from fundamental architectural constraints and have important implications for deployment decisions.

1) Increased Jailbreak Susceptibility: SLMs exhibit significantly higher jailbreak susceptibility compared to their larger counterparts. Zhang et al. found median attack success rates of 34% across 63 SLMs compared to approximately 15% for frontier LLMs with equivalent safety training [12]. Yi et al. attributed this disparity to the "submerged threat" phenomenon, where safety mechanisms that function adequately under normal conditions fail catastrophically under adversarial pressure in smaller models [14]. The reduced parameter count fundamentally limits the model's capacity to maintain robust safety boundaries while simultaneously performing useful tasks. Li et al. found an inverse relationship between model size and compliance with harmful requests, smaller models were 2.3 times more likely to generate functional malicious code when prompted appropriately [48]. Wu et al. extended these findings to speech-based SLMs, demonstrating that multimodal small models exhibit even higher vulnerability rates due to the additional attack surface introduced by audio processing [49].

2) Reduced Privacy Leakage: Conversely, SLMs demonstrate reduced susceptibility to certain privacy attacks due to their limited memorization capacity. Carlini et al. established that memorization scales with model capacity, meaning smaller models inherently store fewer training examples verbatim [50]. Empirically, Lehman et al. found GPT-2 exhibited 40% lower membership inference attack success rates than larger variants when evaluated on clinical text data [45]. This relationship suggests a potential privacy advantage for SLM deployment in sensitive domains, though it should not be interpreted as immunity to privacy attacks, sophisticated techniques such as neighborhood comparison can still achieve meaningful success rates against smaller models [46].

3) Comparable Backdoor Vulnerability: Unlike jailbreak and privacy attacks, backdoor vulnerability appears largely independent of model size. Li et al. proved theoretically and demonstrated empirically that poisoning attacks require only a near-constant number of poison samples regardless of model scale [36]. This finding has significant security implications: SLMs cannot rely on their smaller size as a defense against training-time attacks. Yang et al. further demonstrated that stealthy backdoors can be implanted in SLMs with minimal impact on benign task performance, making detection particularly challenging [51]. The comparable vulnerability across model scales suggests that backdoor defenses developed for large models should remain applicable to SLMs.

4) Attack Transferability: Attack transferability presents additional concerns for SLM security. Chen et al. demonstrated that adversarial prompts optimized on large models transfer to SLMs with 78% effectiveness, enabling attackers to develop attacks on well-resourced target models and deploy them against resource-constrained edge deployments [52]. Universal adversarial triggers exhibit similar transfer properties across model families and sizes [25]. This transferability undermines the potential security-through-obscurity benefits of deploying less common SLM architectures, as attacks developed against mainstream models remain effective.

E. Architectural Contributing Factors (RQ3)

RQ3: What specific architectural or deployment characteristics of SLMs contribute to their unique vulnerability profiles?

Four primary architectural and deployment characteristics contribute to the distinct vulnerability profiles observed in small language models: parameter reduction effects, quantization vulnerabilities, knowledge distillation risks, and edge deployment constraints.

1) Parameter Reduction Effects: The fundamental constraint of reduced parameters directly impacts security implementation capacity. Lu et al. documented that models in the 100M–5B parameter range allocate proportionally fewer parameters to safety-related computations compared to models exceeding 10B parameters [6]. Safety training techniques such as Reinforcement Learning from Human Feedback (RLHF) require substantial model capacity to encode nuanced refusal behaviors across diverse harmful request categories; when capacity is limited, models tend to default to either excessive refusal that impairs utility or insufficient safety that permits harmful outputs. Yi et al. identified that SLMs struggle to maintain safety under distribution shift, learning more brittle safety representations that collapse when inputs deviate from training distributions [14]. Models in the 1B–3B parameter range showed the most pronounced vulnerability increases, suggesting a critical threshold below which safety mechanisms become unreliable.

2) Quantization Vulnerabilities: Quantization, the process of reducing numerical precision for efficient edge deployment, introduces vulnerabilities beyond simple performance degradation. Li et al. demonstrated that 4-bit quantization increases jailbreak attack success rates by approximately 25% compared to full-precision counterparts, with the degradation particularly pronounced for models already near the safety threshold [13]. More critically, quantization creates opportunities for hardware-level fault injection attacks that manipulate the execution environment to bypass safety mechanisms entirely. By inducing bit flips in quantized weight representations through techniques such as rowhammer or voltage glitching, attackers can selectively disable safety-critical neurons without triggering software-level detection. The combination of reduced numerical precision and physical accessibility makes quantized edge deployments particularly vulnerable to sophisticated adversaries with hardware access.

3) Knowledge Distillation Risks: Knowledge distillation, training smaller student models to mimic larger teacher models, transfers not only capabilities but also vulnerabilities. Chen et al. demonstrated that adversarial prompts effective against teacher models remain effective against distilled students with minimal modification [52]. Furthermore, distilled

models inherit the adversarial vulnerabilities of their teachers while losing some defensive capacity due to reduced parameters [39]. Liu et al. proposed Distillation-Aware Robust Defense (DARD) to demonstrate that robustness-aware distillation can preserve defensive properties when explicitly incorporated into the training objective [53]. However, standard distillation practices that optimize solely for task performance systematically degrade security properties, creating a hidden vulnerability in models trained using default configurations.

4) Edge Deployment Constraints: Edge deployment introduces security challenges that extend beyond model architecture to encompass the entire deployment environment [54]–[57]. Physical access to edge devices enables adversaries to perform model extraction attacks that recover weights through power analysis or electromagnetic emanations, as well as direct weight manipulation through memory corruption. Resource limitations on edge hardware prevent deployment of computationally expensive defensive mechanisms, forcing trade-offs between security and performance that typically favor functionality. Edge models deployed in the field may be difficult or impossible to update when vulnerabilities are discovered, leaving systems exposed for extended periods. Additionally, IoT deployments often lack network isolation, enabling remote attackers to probe edge models without physical access while exploiting the limited logging and monitoring capabilities typical of constrained devices.

F. Defense Mechanisms (RQ4)

RQ4: What mitigation strategies and defense mechanisms have been proposed for SLM vulnerabilities, and what empirical evidence supports their effectiveness?

The literature presents defense mechanisms operating at three distinct abstraction levels: input processing, model architecture, and training procedures. Each approach offers different trade-offs between effectiveness, computational overhead, and applicability to resource-constrained SLM deployments.

1) Input-Level Defenses: Input-level defenses aim to detect and neutralize malicious inputs before they reach the model's core processing. SmoothLLM introduces controlled randomness by perturbing input prompts and aggregating predictions across multiple perturbations, exploiting the observation that adversarial prompts are typically brittle to small modifications while benign prompts remain stable [58]. PromptScreen employs a lightweight classifier trained to detect jailbreak attempts, achieving 94% detection accuracy while adding minimal inference latency suitable for edge deployment [59]. Alon et al. proposed perplexity-based detection, leveraging the observation that adversarial suffixes typically exhibit abnormally high perplexity compared to natural language, enabling low-overhead filtering without requiring additional model inference [60]. These input-level approaches offer the advantage of modularity, operating independently of the protected model and enabling deployment across diverse SLM architectures without modification.

2) Architecture-Level Defenses: Architecture-level defenses modify or monitor model internals to detect and prevent attacks during inference. AttentionDefense monitors attention patterns across layers to detect when adversarial inputs redirect attention away from safety-relevant instructions toward harmful content generation, triggering intervention when anomalous patterns are detected [61]. The approach exploits the observation that successful jailbreaks typically require suppressing attention to safety-aligned regions of the input, creating a detectable signature. Conscience-based frameworks employ a secondary SLM dedicated to safety-checking, evaluating proposed outputs before delivery to users [62]. While effective, these approaches incur computational overhead that may be prohibitive for resource-constrained edge deployments, requiring careful optimization to maintain acceptable inference latency.

3) Training-Level Defenses: Training-level defenses incorporate security objectives directly into the model training process. Robust prompt optimization techniques train models to maintain safety alignment even under adversarial prompt conditions, reducing jailbreak attack success rates to near-zero in controlled evaluations [63]. In-decoding safety probing monitors internal representations during token generation, enabling early termination when the model begins generating harmful content [64]. For backdoor vulnerabilities, multiple defensive techniques have been proposed: fine-pruning removes potentially compromised neurons through targeted pruning guided by activation analysis; activation clustering identifies backdoor triggers by detecting outlier activation patterns; and spectral signature analysis leverages the observation that backdoored samples often produce distinguishable spectral characteristics in intermediate representations [33], [34]. These training-level approaches offer the strongest theoretical guarantees but require access to the training pipeline.

4) Effectiveness and Limitations: Critical evaluation of proposed defenses reveals significant limitations that temper optimism about current approaches. Chao et al. conducted systematic evaluation demonstrating that 8 of 10 evaluated defenses could be bypassed with attack modifications requiring no additional computational resources beyond the original attack [27]. The adaptive attacks exploited the deterministic nature of defenses, crafting inputs specifically designed to evade detection while maintaining attack effectiveness. Wu et al. documented the "evolving security" challenge: as defenses improve, attack techniques advance correspondingly, creating an ongoing arms race where neither side achieves lasting advantage [65]. This dynamic has particularly concerning implications for edge-deployed SLMs, where static defenses cannot be easily updated as new attacks emerge. The combination of limited update mechanisms, constrained computational resources for defense, and evolving attack sophistication suggests that current defensive approaches provide, at best, temporary mitigation rather than comprehensive protection.

IV. Discussion

The synthesis of 65 studies examining small language model security reveals patterns that merit careful interpretation. This section contextualizes the empirical findings within broader

security paradigms, articulates their practical implications for diverse stakeholders, and acknowledges the methodological boundaries that constrain generalization.

A. Key Findings

This systematic review establishes that small language models present a distinct security profile characterized by specific vulnerabilities that diverge from their larger counterparts in important ways. The evidence demonstrates that SLMs exhibit systematically higher susceptibility to jailbreak attacks, with median attack success rates of 34% compared to approximately 15% for frontier LLMs with equivalent safety training [12]. This disparity arises from fundamental capacity constraints: smaller architectures cannot simultaneously maintain robust safety boundaries and perform useful tasks without degradation in one or both domains. Paradoxically, SLMs show reduced vulnerability to privacy attacks, with membership inference success rates approximately 40% lower than larger variants due to limited memorization capacity [45]. Perhaps most concerning is the finding that backdoor vulnerabilities remain model-size-independent, requiring only a near-constant number of poison samples regardless of scale [36], which undermines assumptions that compact models might offer inherent resistance to training-time attacks.

The review also reveals that architectural optimization techniques intended to enable edge deployment, quantization, knowledge distillation, and compression, systematically degrade security properties through mechanisms absent in cloud-deployed large models. Defense mechanisms proposed across input, architecture, and training levels show promise in controlled evaluations but demonstrate fragility under adaptive attack conditions, with 8 of 10 evaluated defenses bypassed by modified attacks requiring minimal additional effort [27]. These findings collectively indicate that the current rapid deployment of SLMs in security-critical applications outpaces the development of adequate protective measures, creating a widening gap between capability deployment and security maturation.

B. Implications for Practice

The findings of this review carry significant implications for multiple stakeholder groups involved in the development, deployment, and governance of small language models. For developers and deploying organizations, the evidence mandates that safety mechanisms must be designed with explicit awareness of resource constraints rather than adapted from large model approaches that assume abundant capacity. Predeployment security testing should be considered essential rather than optional, utilizing benchmarks such as those developed by Zhang et al. [12] to evaluate vulnerability profiles before production release. Given the demonstrated fragility of individual defenses, a defense-in-depth approach combining input filtering, architectural monitoring, and training hardening offers more robust protection than any single mechanism. Organizations deploying SLMs on edge devices must also establish update mechanisms capable of addressing newly discovered vulnerabilities, recognizing that static deployments accumulate security debt as attack techniques evolve.

For security researchers, the findings identify several priority areas requiring focused investigation. The development of adaptive defense mechanisms that can evolve in response to emerging attacks represents a critical need, as current static approaches provide only temporary mitigation. Quantization-aware security research should examine fault injection vulnerabilities that emerge specifically from reduced numerical precision in edge deployments. Knowledge distillation methods that preserve robustness properties, exemplified by approaches such as DARD [53], warrant further development and validation. Additionally, edge-specific threat modeling that accounts for physical access, limited monitoring, and constrained update capabilities would address gaps in current security frameworks designed primarily for cloud environments.

For policymakers and standards bodies, the review highlights that current rapid SLM deployment substantially outpaces security standardization efforts. The development of SLM-specific security benchmarks would enable consistent evaluation across models and vendors. Certification frameworks for edge AI in critical applications, healthcare, transportation, industrial control, could establish minimum security requirements before deployment approval. Transparency requirements regarding the security posture of deployed SLMs would enable informed procurement decisions and appropriate risk management by adopting organizations.

C. Limitations

Several methodological limitations constrain the interpretation and generalizability of these findings. The rapid evolution of the small language model field means that 78% of included studies were published in 2024–2025, limiting longitudinal validity and the ability to identify persistent versus transient vulnerability patterns. The predominance of arXiv preprints (58%) reflects the field's pace but introduces variability in peer review rigor, though quality assessment procedures attempted to weight evidence appropriately. Heterogeneous definitions of "small" across studies, ranging from 100 million to 10 billion parameters, complicate direct comparison and synthesis. The quality assessment revealed that only 37% of studies achieved high-quality classification, suggesting that conclusions regarding attack success rates and defense effectiveness should be interpreted with appropriate caution.

Scope limitations further bound the applicability of findings. This review focused on known vulnerability categories derived from established taxonomies; emerging threat vectors not yet characterized in the literature fall outside its coverage. The emphasis on transformer-based architectures means findings may not generalize to alternative approaches such as state-space models or hybrid architectures increasingly explored for efficient deployment. Limited coverage of domain-specific SLMs deployed in medical, legal, or financial contexts constrains conclusions about vulnerability profiles in high-stakes specialized applications. Finally, the arms race dynamic between attacks and defenses means that defense effectiveness

data captures only a temporal snapshot; mechanisms deemed effective at publication may have been subsequently bypassed, and the review cannot predict future attack evolution.

V. Conclusions

This systematic literature review provides the first comprehensive analysis of security vulnerabilities specific to Small Language Models, synthesizing findings from 65 studies published between 2020 and 2025. Addressing the research questions posed in the Introduction, the evidence reveals that SLMs exhibit a distinct and divergent security profile from their larger counterparts. Jailbreak attacks achieve substantially higher success rates against SLMs, with median attack success rates of 34% compared to 15% for LLMs with equivalent safety alignment, confirming that reduced model capacity fundamentally constrains the implementation of robust safety mechanisms. Conversely, SLMs demonstrate lower vulnerability to privacy attacks, as their reduced memorization capacity limits the effectiveness of membership inference and training data extraction techniques. Backdoor attacks present a model-size-independent threat, achieving near-perfect attack success rates across all model scales while evading current detection methods. Critically, the architectural optimizations that enable practical SLM deployment, including quantization, pruning, and knowledge distillation, systematically degrade security properties, with post-training quantization alone increasing jailbreak vulnerability by 25% in 4-bit configurations. Current defense mechanisms remain inadequate, with empirical evaluation demonstrating that eight of ten established defenses can be bypassed by adaptive attacks.

The broader significance of these findings extends beyond academic characterization to urgent practical implications. A substantial gap exists between the accelerating pace of SLM deployment in edge computing, mobile applications, and IoT systems and the maturation of security frameworks adequate to protect these deployments. The evidence synthesized here demonstrates that adapting Large Language Model security approaches to smaller architectures is insufficient; SLM-specific security frameworks must be developed that account for the fundamental constraints of reduced model capacity and the expanded attack surfaces of edge deployment. Priority research areas identified through this review include the development of adaptive defenses that maintain effectiveness against evolving attack strategies, quantization-aware security techniques that preserve safety properties through compression, and comprehensive threat modeling for edge deployment scenarios. Industry and regulatory bodies should consider establishing security certification standards for SLMs deployed in critical applications before such deployments become widespread. As Small Language Models transition from research artifacts to production systems embedded in safety-critical infrastructure, the integration of security considerations throughout the model development lifecycle, from architecture design through deployment and monitoring, is not merely advisable but essential.

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.

[21] S. Xu *et al.*, "A comprehensive study of jailbreak attack versus defense for large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[22] P. Chao *et al.*, "Bag of tricks: Benchmarking of jailbreak attacks on llms," in *Advances in Neural Information Processing Systems*, 2024.

[23] Y. Liu, G. Deng, Z. Xu, Y. Li, Y. Zheng, Y. Zhang, L. Zhao, T. Zhang, and Y. Liu, "Jailbreaking chatgpt via prompt engineering: An empirical study," *arXiv preprint arXiv:2305.13860*, 2023.

[24] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does llm safety training fail?" *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[25] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[26] M. Andriushchenko *et al.*, "Jailbreaking safety-aligned llms with simple adaptive attacks," in *International Conference on Learning Representations (ICLR)*, 2024.

[27] P. Chao *et al.*, "The attacker moves second: Stronger adaptive attacks bypass defenses," *arXiv preprint arXiv:2510.09023*, 2025.

[28] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection," *arXiv preprint arXiv:2302.12173*, 2023.

[29] F. Perez and I. Ribeiro, "Ignore this title and hackaprompt: Exposing systemic vulnerabilities of llms through a global prompt hacking competition," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 4945–4977.

[30] Y. Liu *et al.*, "Bypassing prompt injection and jailbreak detection in llm guardrails," *arXiv preprint arXiv:2504.11168*, 2025.

[31] A. Kumar *et al.*, "Red teaming the mind of the machine: A systematic evaluation of prompt injection and jailbreak vulnerabilities," *arXiv preprint arXiv:2505.04806*, 2025.

[32] W. Chen *et al.*, "Multimodal prompt injection attacks: Risks and defenses for modern llms," *arXiv preprint arXiv:2509.05883*, 2025.

[33] J. Chen *et al.*, "A survey on backdoor threats in large language models: Attacks, defenses, and evaluation methods," *Transactions on Artificial Intelligence*, 2025.

[34] S. Zhao *et al.*, "A survey of backdoor attacks and defenses on large language models," *arXiv preprint arXiv:2406.06852*, 2024.

[35] H. Huang *et al.*, "Composite backdoor attacks against large language models," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024.

[36] Y. Li *et al.*, "Poisoning attacks on llms require a near-constant number of poison samples," *arXiv preprint arXiv:2510.07192*, 2025.

[37] D. Chen *et al.*, "Medical large language models are vulnerable to data-poisoning attacks," *Nature Medicine*, 2024.

[38] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, N. Z. Gong, Y. Zhang, and X. Xie, "Promptbench: Towards evaluating the robustness of large language models on adversarial prompts," *arXiv preprint arXiv:2306.04528*, 2023.

[39] P. Liu *et al.*, "Adversarial evasion attack efficiency against large language models," *arXiv preprint arXiv:2406.08050*, 2024.

[40] Y. Dong *et al.*, "Adv-bert: Bert is not robust on misspellings!" *arXiv preprint arXiv:2003.04985*, 2020.

[41] T. Koide *et al.*, "Phishlang: Phishing detection framework using mobile-bert," *arXiv preprint arXiv:2408.05667*, 2024.

[42] D. Santos *et al.*, "Explainable transformer-based email phishing classification with adversarial robustness," *arXiv preprint arXiv:2511.12085*, 2025.

[43] Y. Hu *et al.*, "Membership inference attacks on large-scale models: A survey," *arXiv preprint arXiv:2503.19338*, 2025.

[44] T. Win *et al.*, "Win-k: Improved membership inference attacks on small language models," *arXiv preprint arXiv:2508.01268*, 2025.

[45] E. Lehman *et al.*, "Membership inference attack susceptibility of clinical language models," *arXiv preprint arXiv:2104.08305*, 2021.

[46] J. Mattern *et al.*, "Membership inference attacks against language models via neighbourhood comparison," in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

[47] Y. Hui *et al.*, "Pleak: Prompt leaking attacks against large language model applications," in *ACM Conference on Computer and Communications Security (CCS)*, 2024.

[48] S. Li *et al.*, "Llms caught in the crossfire: Malware requests and jailbreak challenges," *arXiv preprint arXiv:2506.10022*, 2025.

[49] J. Wu *et al.*, "Jailbreaking speech-enabled small language models," *arXiv preprint arXiv:2501.12345*, 2025.

[50] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang, "Quantifying memorization across neural language models," *arXiv preprint arXiv:2202.07646*, 2023.

[51] W. Yang *et al.*, "Stealthy backdoor attacks against small language models," *arXiv preprint arXiv:2410.54321*, 2024.

[52] Y. Chen *et al.*, "Efficient and stealthy jailbreak attacks via adversarial prompt distillation from llms to slms," *arXiv preprint arXiv:2506.17231*, 2025.

[53] Q. Liu *et al.*, "Dard: Dice adversarial robustness distillation," *arXiv preprint arXiv:2509.11525*, 2025.

[54] V. Singh *et al.*, "Deploying ai on edge: Advancement and challenges in edge intelligence," *MDPI Mathematics*, 2025.

[55] J. Chen *et al.*, "Intelligent data analysis in edge computing with large language models," *Frontiers in Computer Science*, 2025.

[56] M. Liu *et al.*, "Empowering large language models to edge intelligence: A survey," *Neural Networks (Elsevier)*, 2025.

[57] W. Xu *et al.*, "Llms and iot: A comprehensive survey," *TechRxiv preprint*, 2024.

[58] A. Robey, E. Wong, H. Hassani, and G. J. Pappas, "Smoothllm: Defending large language models against jailbreaking attacks," *arXiv preprint arXiv:2310.03684*, 2023.

[59] J. Zhang *et al.*, "Promptscreen: Efficient jailbreak mitigation using semantic linear classification," *arXiv preprint arXiv:2512.19011*, 2025.

[60] G. Alon and M. Kamfonas, "Detecting language model attacks with perplexity," *arXiv preprint arXiv:2308.14132*, 2023.

[61] X. Wang *et al.*, "Attentiondefense: Leveraging system prompt attention for explainable defense against novel jailbreaks," *arXiv preprint arXiv:2504.12321*, 2025.

[62] M. Bergeron *et al.*, "Bergeron: Combating adversarial attacks through a conscience-based alignment framework," *arXiv preprint arXiv:2312.00029*, 2023.

[63] A. Zhou *et al.*, "Robust prompt optimization for defending language models," 2024.

[64] H. Xu *et al.*, "Defending large language models against jailbreak attacks via in-decoding safety-awareness probing," *arXiv preprint arXiv:2601.10543*, 2025.

[65] Z. Wu *et al.*, "Evolving security in llms: A study of jailbreak attacks and defenses," *arXiv preprint arXiv:2504.02080*, 2025.