On-Premise SLMs vs. Commercial LLMs: Prompt Engineering and Incident Classification in SOCs and CSIRTs

Gefté Almeida[1], **Marcio Pohlmann**[1], **Alex Severo**[1]
Diego Kreutz[1], **Tiago Heinrich**[2], **Lourenço Pereira**[3]

[1] AI Horizon Labs – PPGES – Universidade Federal do Pampa (UNIPAMPA)
[2]Max Planck Institute for Informatics (MPI)
[3]Instituto Tecnológico de Aeronáutica (ITA)

{geftealmeida,marciopohlmann,alexsevero}.aluno@unipampa.edu.br

diegokreutz@unipampa.eud.br, theinric@mpi-inf.mpg.de, ljr@ita.br

Abstract. *In this study, we evaluate open-source models for security incident classification, comparing them with proprietary models. We utilize a dataset of anonymized real incidents, categorized according to the NIST SP 800-61r3 taxonomy and processed using five prompt-engineering techniques (PHP, SHP, HTP, PRP, and ZSL). The results indicate that, although proprietary models still exhibit higher accuracy, locally deployed open-source models provide advantages in privacy, cost-effectiveness, and data sovereignty.*

1. Introduction

According to CERT.br, Brazil reported over 516k security incidents in 2024 and more than 181k in the first half of 2025, underscoring a persistent upward trend that challenges SOCs and CSIRTs to manage high alert volumes efficiently[1]. To alleviate this overload, AI-driven solutions, particularly prompt-engineering techniques such as Progressive Hint Prompting (PHP), have demonstrated over 90% accuracy with models like GPT-4o and Gemini 2 [Severo et al. 2025a]. However, the use of commercial LLMs involves high costs, dependence on external providers, and privacy risks imposed by the General Personal Data Protection Law (LGPD). As an alternative, locally executed *open-source* models offer greater control, privacy, and autonomy, while maintaining competitive performance on accessible *hardware* and reducing operational costs.

Given these limitations and the need to explore open alternatives, we perform an empirical evaluation of *open-source* models using a balanced sample of anonymized real incidents, applying the same prompt-engineering techniques Progressive Hint Prompting (PHP), Self-Hint Prompting (SHP), Hypothesis Testing Prompting (HTP) and the NIST SP 800-61r3 taxonomy (see Table 1), following the previous study conducted with online LLMs [Severo et al. 2025a]. In the evaluation, two groups of models were compared (Table 2 and Table 3)[2]: the first group composed of smaller models, close to *Llama-3.1-FoundationAI-SecurityLLM-Base-8B* [Kassianik et al. 2025], trained specifically for the cybersecurity domain; and the second group composed of larger models, around 70B parameters, representing the most robust open variants within their respective families. Two

[1] https://stats.cert.br/incidentes/

[2] https://github.com/AILabs4All/FrameworkPE/blob/main/paper/metadata_apendice.md

models from each family were selected, enabling comparison across different architectures and scales within the evaluated set.

2. State of the Art

The efficacy of Large Language Models (LLMs) in classifying security incidents using structured taxonomies, such as NIST SP 800-61r3, relies heavily on domain-specific knowledge and contextual reasoning [Salahuddin et al. 2025, Kassianik et al. 2025]. General-purpose models often fail to capture cybersecurity-specific nuances, which motivates the adoption of *Domain-Adaptive Continuous Pretraining* (DAP) methodologies. Examples such as *Foundation-Sec-8B* demonstrate that continuous pretraining on specialized corpora yields substantial gains in benchmarks like CTIBench, while also improving typical SOC-related tasks including triage, alert enrichment, and TTP extraction [Kassianik et al. 2025, Tellache et al. 2024].

Table 1. Security Incident Categorization based on NIST SP 800-61r3

Code	Category	Description	Priority
CAT1	Account Compromise	Unauthorized access to user or administrator accounts.	5
CAT2	Malware	Infection by malicious code compromising devices or data.	5
CAT3	Denial-of-Service Attack (DoS/DDoS)	Making systems or networks unavailable.	4
CAT4	Data Exfiltration or Leakage	Unauthorized access, copying, or disclosure of sensitive data.	5
CAT5	Vulnerability Exploitation	Use of known or unknown flaws to compromise assets.	5
CAT6	Insider Abuse	Intentional or negligent actions by internal users.	5
CAT7	Social Engineering	Deceiving people to obtain access or information.	3
CAT8	Physical or Infrastructure Incident	Physical breach impacting computational assets.	4
CAT9	Unauthorized Modification	Unauthorized changes to systems, data, or configurations.	3
CAT10	Misuse of Resources	Unauthorized use of systems for other purposes.	2
CAT11	Vendor/Third-Party Problem	Incident originating from a third-party security failure.	4
CAT12	Intrusion Attempt	Hostile attempts to break in not yet confirmed as successful.	3

In parallel, the study by Irugalbandara *et al.* [Irugalbandara et al. 2024] evaluated nine *open-source* models and twenty-nine quantized variants, showing that self-hosted solutions can reduce costs by factors ranging from 5× to 29×, while maintaining latency equal to or lower than proprietary models and offering greater operational stability. Whereas GPT-4 operates at an approximate cost of US$ 0.09 per thousand *tokens*, local models range between US$ 0.003 and US$ 0.018 per thousand *tokens*, reinforcing the feasibility of large-scale adoption.

These findings contrast directly with the use of proprietary LLMs (such as GPT-4o and Claude), whose dependency on external providers introduces risks related to privacy, legal compliance (e.g., LGPD), and data sovereignty [Pan and Wang 2025, Noreika 2025]. Conversely, *open-source* models (such as *Llama*, *Mistral*, and *Qwen*) offer greater operational control, transparency, and flexibility for fine-tuning capabilities that are essential in environments handling sensitive information. Total Cost of Ownership (TCO) analyses indicate that *on-premise* deployments become economically advantageous after short break-even periods, particularly for small and medium-sized models [Pan and Wang 2025]. Thus, open-source models emerge as viable and strategically attractive alternatives for corporate and institutional scenarios requiring high levels of privacy, autonomy, and auditability.

3. *Pipeline* for Automated Classification with SLMs

The methodological architecture adopted in this work is based on a modular automated classification pipeline composed of five stages: Input Data, Preprocessing, Processing, Analysis, and Output, as illustrated in Figure 1. This structure is inspired by the framework introduced in previous work [Severo et al. 2025a], but it has been fully adapted for local execution, enabling the assessment of *open-source* models in real-world scenarios of security incident categorization.

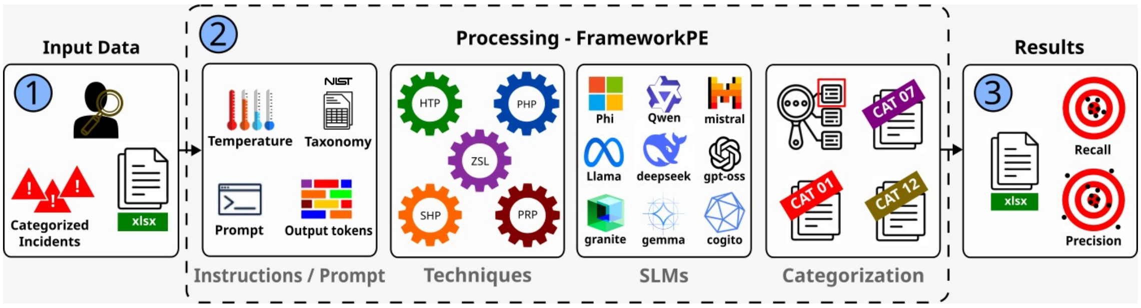

Figure 1. Flowchart of the automated classification pipeline using SLMs.

Input Data. The incident reports used in this study are the same as those employed in the original experiment with LLMs. The dataset consists of a balanced sample of 24 security incidents selected from the 194 records used in [Severo et al. 2025a]. All records were previously anonymized using the AnonLFI tool[3], ensuring the removal of names, IP addresses, and other sensitive identifiers. The reference labels (*ground truth*) were preserved exactly as defined by two cybersecurity specialists in the prior study, serving as the baseline for comparative evaluation of the open-source models.

Preprocessing. The preprocessing stage maintained the same semantic structure adopted in the original pipeline, but was adapted to the formats required by *open-source* models. Input and output *token* limits were defined, and a uniform temperature[4] was applied across all inferences. The entire process was automated through Python scripts, ensuring reproducibility and traceability across runs.

Processing. The processing stage forms the core of the pipeline and applies the five prompt-engineering strategies (PHP, SHP, HTP, PRP, and ZSL) to multiple models executed locally using the Ollama *framework* in a controlled GPU-enabled environment. To ensure fair comparison across models, inference parameters such as temperature, maximum number of *tokens*, and recursion depth were standardized. Additionally, operational metrics including average execution time, memory consumption, and response length were collected to assess efficiency and classification robustness.

The evaluated models span a broad architectural spectrum, ranging from 7B to 70B parameters and covering different *open-source* families, attention mechanisms, activation functions, and normalization strategies. For clarity and reproducibility, their main architectural characteristics are organized into two groups. Group 1 contains larger or

[3] `https://github.com/gt-rnp-lfi/anon`

[4] In another work we do an extensive evaluation of the effect of temperature in Small Languages Models (SLMs)[Pohlmann et al. 2025].

mid-sized models focused on general-purpose reasoning, while Group 2 includes compact and efficiency-oriented models optimized for local inference. Detailed specifications for both groups are presented in Tables 2 and 3.

Table 2. Group 1 models and their main architectural characteristics.

Model	Size	Context Window (tokens)	Attention	Activation	Normalization
Cogito 70B	70B	128,000	MHA	SwiGLU	RMSNorm
DeepSeek R1 70B	70B	128,000	MLA	SwiGLU	RMSNorm
Falcon3 10B	10B	32,000	GQA	ReLU	LayerNorm
Gemma 2 27B	27B	8,192	GQA	GeGLU	RMSNorm
Gemma 3 27B	27B	128,000	GQA	GeGLU	RMSNorm
GPT-OSS 20B	20B	131,072	MoE	SwiGLU	RMSNorm
Llama 3.3 70B	70B	131,072	GQA	SwiGLU	RMSNorm
Mistral Small 24B	24B	32,768	GQA	SwiGLU	RMSNorm
Phi-4 14B	14B	16,000	MHA	GeGLU	RMSNorm
Qwen2.5 32B	32B	131,072	GQA	SwiGLU	RMSNorm
Qwen3 32B	32B	131,072	GQA	SwiGLU	RMSNorm

Table 3. Group 2 Models and their main architectural characteristics.

Model	Size	Context Window (tokens)	Attention	Activation	Normalization
Qwen3 8B	8B	262,144	GQA	SwiGLU	RMSNorm
Qwen2.5 7B	7B	131,072	GQA	SwiGLU	RMSNorm
Cogito 8B	8B	128,000	MHA	SwiGLU	RMSNorm
DeepSeek R1 8B	8B	128,000	MLA	SwiGLU	RMSNorm
Falcon3 7B	7B	32,000	GQA	ReLU	LayerNorm
Gemma 2 9B	9B	8,192	GQA	GeGLU	RMSNorm
Gemma 3 12B	12B	128,000	GQA	GeGLU	RMSNorm
Granite3.2 8B	8B	131,072	MHA	SwiGLU	RMSNorm
Llama 3.1 8B	8B	128,000	MHA	SwiGLU	RMSNorm
Mistral 7B	7B	32,768	GQA, SWA	SwiGLU	RMSNorm
Foundation-Sec 8B	8B	128,000	MHA	SwiGLU	RMSNorm

4. Results and Discussion

The results derived from the models were benchmarked against the baseline established in the prior study [Severo et al. 2025b, Severo et al. 2025a], which utilized proprietary models, evaluating both accuracy and operational efficiency. The evaluation involved two primary metrics: classification accuracy, measured by the correspondence between the category assigned by the model and the expert-defined *ground truth* and computational efficiency, assessed through average inference time and GPU usage.

4.1. Comparison Across Models and Techniques

Figures 2 and 3 present a comparison of the performance of the five prompt-engineering strategies (PHP, SHP, PRP, HTP, and ZSL) applied to both groups of models. It is evident that the approaches based on progressive hints (PHP and SHP) continue to exhibit the best overall performance, achieving higher and more consistent accuracy rates across the evaluated models. This result aligns with the pattern identified in [Severo et al. 2025a] and reinforces the notion that incremental hinting facilitates semantic alignment between the model and the categorization instructions, contributing to improved contextual generalization.

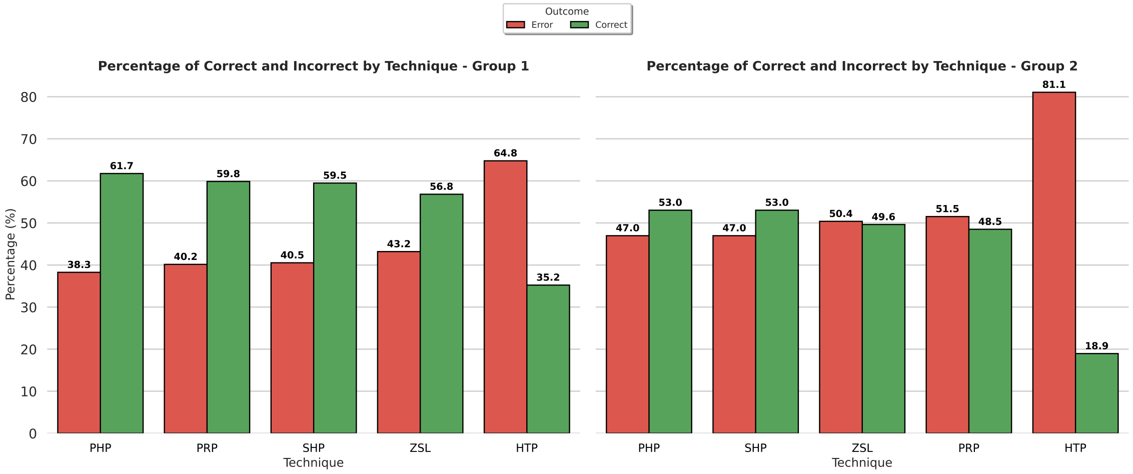

Figure 2. Percentage of correct and incorrect predictions by prompt technique.

In Group 1, composed predominantly of large models (between 20B and 70B parameters), *Progressive Hint Prompting* (PHP) achieved the best average performance, with an accuracy of 61.7%, followed by PRP and SHP, both in the 59–60% range. *Zero-Shot Learning* (ZSL) obtained an intermediate result (56.8%), whereas *Hypothesis Testing Prompting* (HTP) exhibited significantly lower performance (35.2%). The gap of approximately 26 percentage points between PHP and HTP highlights the importance of iterative refinement and the gradual decomposition of reasoning in higher-capacity models. Additionally, architectures featuring optimized attention mechanisms (such as GQA and MoE) and modern activation functions (SwiGLU, GeGLU) showed better responsiveness to progressive *prompting* techniques.

In Group 2, composed of smaller models (between 7B and 12B parameters), a general reduction in accuracy levels was observed, although the superior performance pattern of PHP and SHP persisted, with both achieving averages close to 53%. Progressive Response Prompting (PRP) and ZSL displayed similar performance (49–51%), while HTP was again the least effective method, reaching only 18.9% accuracy. This more pronounced decline in techniques requiring complex reasoning indicates that smaller-scale models are more sensitive to the cognitive load of the instructions and degrade more rapidly when the *prompt* demands hypothetical or chained inferences.

In summary, the comparison between the groups shows that both model size and attention architecture directly influence the stability of *prompting* techniques. Models with optimized attention mechanisms, such as GQA and MoE, as well as those with larger context windows, exhibited greater consistency across the evaluated techniques, whereas models relying on conventional attention (MHA) or simpler activation functions (ReLU) showed higher performance variability. These findings indicate that architectural *design* and the choice of *prompting* strategy interact in a decisive manner in determining overall results, with progressive approaches (PHP and SHP) providing the greatest robustness even under different model configurations.

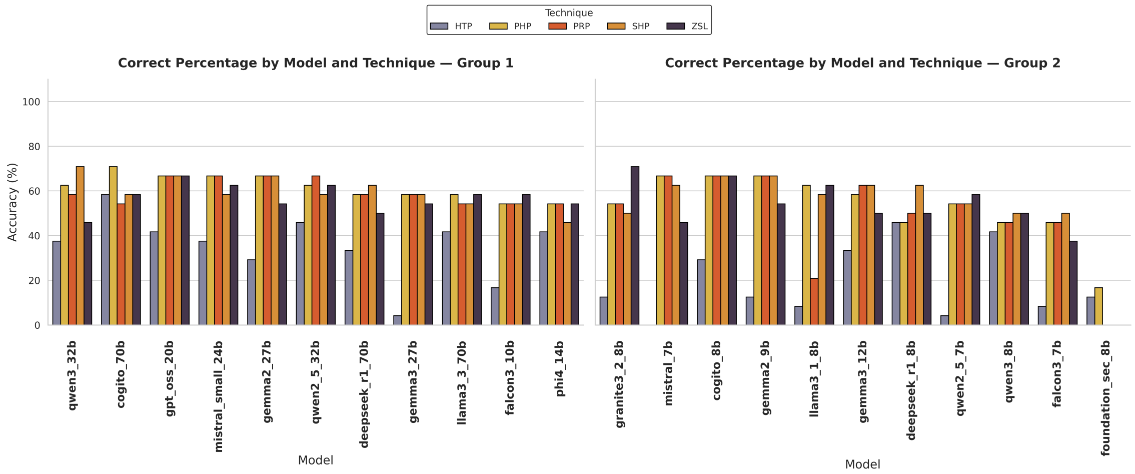

Figure 3. Percentage of correct and incorrect predictions by Model × Prompt Technique.

5. Final Remarks

This study indicates that SLMs are a technically viable alternative for automated classification of security incidents, particularly in institutional environments that require data privacy, cost predictability, and operational autonomy. Although open-source models between 8B and 20B parameters reached accuracies around 60%, below the levels above 90% typically achieved by LLMs, they demonstrated stability and semantic consistency for initial triage and decision-support tasks in SOCs and CSIRTs. Prompting strategies such as *Progressive-Hint Prompting* and *Self-Hint Prompting* mitigated context limitations inherent to smaller models, while the use of the Ollama framework ensured full control over data, compliance with the LGPD, and predictable operational costs.

For future work, we plan to extend the dataset with incidents from multiple sectors and languages, and evaluate *fine-tuning* techniques such as LoRA to improve model performance in specific cybersecurity domains. We also intend to incorporate more granular metrics, including *precision*, *recall* and *F1-score*, implement a continuous-learning pipeline to reassess models as new incidents arise, and develop a visual interface for SOC and CSIRT workflows with emphasis on explainability and traceability. These advances aim to consolidate SLMs as autonomous, interpretable and auditable solutions, strengthening digital sovereignty and cyber intelligence in critical environments.

Acknowledgments

This research was partially supported by the CNPq[5], grant 409743/2025-9; by CAPES[6], under Financing Code 001; and by FAPERGS[7], through grant agreements 24/2551-0001368-7 and 24/2551-0000726-1.

References

[Irugalbandara et al. 2024] Irugalbandara, C., Mahendra, A., Daynauth, R., Arachchige, T. K., Dantanarayana, J., Flautner, K., Tang, L., Kang, Y., and Mars, J. (2024). Scaling

[5] https://www.gov.br/cnpq/pt-br

[6] https://www.gov.br/capes/pt-br

[7] https://fapergs.rs.gov.br

down to scale up: A cost-benefit analysis of replacing openai's llm with open source slms in production. *arXiv preprint arXiv:2312.14972*.

[Kassianik et al. 2025] Kassianik, P., Saglam, B., Chen, A., Nelson, B., Vellore, A., Aufiero, M., Burch, F., Kedia, D., Zohary, A., Weerawardhena, S., Priyanshu, A., Swanda, A., Chang, A., Anderson, H., Oshiba, K., Santos, O., Singer, Y., and Karbasi, A. (2025). Llama-3.1-FoundationAI-SecurityLLM-Base-8B Technical Report. Technical Report.

[Noreika 2025] Noreika, A. (2025). Open Source vs Proprietary LLMs: The Key Differences. *SentiSight (Neurotechnology)*. Online Article.

[Pan and Wang 2025] Pan, G. and Wang, H. (2025). A Cost-Benefit Analysis of On-Premise Large Language Model Deployment: Breaking Even with Commercial LLM Services. Working Paper, Carnegie Mellon University.

[Pohlmann et al. 2025] Pohlmann, M., Severo, A., Almeida, G., Kreutz, D., Heinrich, T., and Pereira, L. (2025). Temperature in SLMs: Impact on incident categorization in on-premises environments. `https://arxiv.org/abs/2511.19464`.

[Salahuddin et al. 2025] Salahuddin, S., Hussain, A., Löppönen, J., Jutila, T., and Papadimitratos, P. (2025). Less Data, More Security: Advancing Cybersecurity LLMs Specialization via Resource-Efficient Domain-Adaptive Continuous Pre-training with Minimal Tokens. *arXiv preprint arXiv:2412.01633*.

[Severo et al. 2025a] Severo, A., Kreutz, D., Bertholdo, L., and Lautert, D. (2025a). Framework de classificação automatizada de incidentes de segurança com llms utilizando técnicas de engenharia de prompt. In *Anais do SBSeg*.

[Severo et al. 2025b] Severo, A., Lautert, D., Almeida, G., Kreutz, D., Rodrigo, G., Jr, L. P., and Bertholdo, L. (2025b). LLMs e engenharia de prompt para classificação automatizada de incidentes em SOCs. In *Anais Estendidos do XXV SBSeg*. SBC.

[Tellache et al. 2024] Tellache, A., Korba, A. A., Mokhtari, A., Moldovan, H., and Ghamri-Doudane, Y. (2024). Advancing Autonomous Incident Response: Leveraging LLMs and Cyber Threat Intelligence. *Preprint*.