Prompt Ward: LLM-Enabled Sensitive-data Handling for Community Services in Smart Cities

Milena Favale
Dept. of Economic Sciences
University of Salento
Lecce, Italy

Ali Aghazadeh Ardebili
CRISR Research Center, Dept. of Economic Sciences
University of Salento
Lecce, Italy

Marco Zappatore
Dept. of Economic Sciences
University of Salento
Lecce, Italy

Antonella Longo
Data Lab, Dept. of Economic Sciences
University of Salento
Lecce, Italy

Antonio Ficarella
Green Engine Lab, CRISR Research Center, University of Salento
Lecce, Italy

***Abstract*—In smart city and community ecosystems, generative AI (GenAI) has emerged as a transformative technology. For instance, public administrations (PAs) and community service providers are increasingly adopting conversational agents such as OpenAI's ChatGPT and Google DeepMind's Gemini. But uncontrolled use of public chatbots creates practical exposure pathways for proprietary information and sensitive personal or organizational data. This paper presents PromptWard, a computationally lightweight, on-premises browser add-in that intercepts prompts, detects personally identifiable information (PII) and other sensitive entities, and masks them before content is submitted to external LLM services. PromptWard couples a content script for real-time input capture with a background service worker that applies prompt-based redaction and validates sanitized outputs. We empirically evaluate PromptWard using three LLMs (llama3-8b-8192, gemma2-9b-it, and compound-beta) across two web platforms (ChatGPT and HumanizeAI.pro) on a 30-input, six-category PII dataset with increasing difficulty. gemma2-9b-it achieves the best overall accuracy (57/60) with stable latency (mean 492 ms). Extended validation (110 cases) yields precision 0.84, recall 0.87, and F1-score 0.85, while resisting 8/10 prompt-injection attempts. Remaining limitations include partial masking of specific formats and over-sensitivity to ambiguous PII-like strings, suggesting targets for refinement.**

***Index Terms*—LLM for Cybersecurity, Samrt Cities and Community Services , Shadow AI, Data Exposure, Prompt Security, Privacy Risks, Compliance, Generative AI Governance, Explainable AI, Data Leakage.**

I. Introduction

LLMs are rapidly embedded into organisational workflows for drafting, summarisation, search and customer-facing interactions [1]–[3]. This uptake is accompanied by a measurable rise in policy violations involving sensitive data shared with GenAI tools, turning employee use into a practical governance and security concern rather than a purely theoretical risk [4], [5]. Staff may unintentionally include proprietary content, credentials, regulated personal data or internal incident details in prompts and follow-up messages [6]–[8], particularly in smart-city settings where municipal offices, service providers and community staff routinely handle citizen-related data.

PromptWard is proposed as a computationally lightweight, on-premises safeguarding tool addressing two research questions: (RQ1) to what extent can a lightweight, low-code browser add-in make privacy-preserving AI interactions accessible to non-technical users in small organisations and public administrations; and (RQ2) can prompt-based, on-premises PII detection and masking effectively reduce exposure of sensitive personal, financial or health-related information during interactions with public LLM chatbots. Its primary contribution is a lightweight privacy-preserving software layer enabling safer conversational assistance in organisational environments.

II. Proposed Solution: PromptWard

PromptWard is a low-code browser extension composed of three modules: an Integrated Safety Classifier that filters dangerous or non-compliant prompts and outputs in real time; a PII Detection Module offering predefined and extensible rules for personal data and regulated terms (e.g. GDPR); and a Policy Enforcement component allowing rules such as blocking, warning or anonymising content. Architecturally, PromptWard couples an Input Monitoring Layer (content script) with a Processing and Redaction Layer (background service worker), communicating through asynchronous message passing.

The content script uses `checkTextBox` and `userIsTyping` to detect typing in input fields, text areas and content-editable elements, extracts the current text via `getText`, and sends it to the background worker via `askBG`. The background worker (`hidePII`) formats explicit instructions specifying the PII categories to detect, including names, emails, phone numbers, addresses, birth dates and places, card numbers, IBANs and tax/social-security IDs, and instructs the LLM to return only the redacted text with detected entities replaced by a placeholder. The worker validates the response against failure patterns before returning it via `sendReply`; if no PII is found, the original text is returned unchanged, and if the LLM request fails, the original unmasked text is passed through to avoid data loss. This design keeps the two layers independently updatable and

TABLE I
OVERALL COMPARATIVE RESULTS (60 ATTEMPTS PER MODEL)

Model	Accuracy	Mean latency (ms)	Range (ms)
llama3-8b-8192	53/60	534	–
gemma2-9b-it	57/60	492	362–870
compound-beta	53/60	1147	up to 3953

relies on standard Chrome Manifest V3 features, so detection instructions or the underlying model can be changed easly.

III. EVALUATION

PromptWard was evaluated using three LLMs accessed via the Groq API (llama3-8b-8192, gemma2-9b-it, compound-beta) across two platforms, ChatGPT and HumanizeAI.pro, on a 30-input, six-category PII dataset with five difficulty levels each. Table I summarises overall accuracy and latency.

Gemma2-9b-it achieved the highest accuracy, failing only on obfuscated email delimiters and international phone prefixes, while offering the most stable latency. llama3-8b-8192 showed partial masking on international phone formats and address civic numbers. compound-beta matched llama3's accuracy but with substantially higher and more variable latency, and its large context window offered no advantage for short-text detection. All models correctly handled obfuscated patterns such as bracket-substituted email symbols, confirming that prompt-based detection addresses evasion techniques that defeat simple regex approaches, although international phone formats remained a shared weakness.

Based on these results, gemma2-9b-it was selected for extended validation on 110 cases (60 standard PII inputs, 10 prompt-injection attempts, 20 non-PII inputs, 20 ambiguous cases). This yielded precision 0.84, recall 0.87 and F1-score 0.85 (TP=61, TN=28, FP=12, FN=9), a mean latency of 484 ms (SD 133 ms), and correct handling of 8 of 10 prompt-injection attempts. The system showed a zero false-positive rate on genuine non-PII inputs but a 60% false-positive rate on deliberately ambiguous inputs, a trade-off favouring privacy protection at the cost of occasional unnecessary masking.

IV. DISCUSSION AND LIMITATIONS

PromptWard appears suitable as a lightweight safeguard for organisations and community services in smart cities, particularly where IT skills are scarce, since its interaction model resembles familiar everyday app usage rather than requiring specialised infrastructure. It may also offer smaller organisations a lower-cost alternative to local LLM deployment while still enabling use of external providers.

Several limitations remain. Detection depends entirely on the underlying LLM's behaviour, which cannot fully replicate the precision of rule-based (regex) systems, and no tested model guaranteed complete masking in every context. The architecture introduces latency from the round trip between browser and remote LLM endpoint, creating a residual risk that unmasked text could be submitted if a user acts before masking completes. Fallback logic returns unmasked text on request failure, which protects user input from being lost but cannot guarantee consistent masking under all conditions. The add-in also only monitors standard input, textarea and content-editable elements, so PII inserted via third-party scripts or auto-filled forms could bypass monitoring. Finally, PromptWard has not yet been tested in a real municipal or e-government deployment.

V. CONCLUSION AND FUTURE WORK

PromptWard demonstrates that a lightweight, browser-level privacy layer can meaningfully reduce PII exposure in interactions with public LLM chatbots without requiring workflow changes, reliably masking well-structured entities such as emails and IBANs while showing more variable performance on locations and initials. Future work includes introducing a backend (e.g. FastAPI) to manage requests and fallback logic more robustly, adding prompt-injection detection modules, supporting configurable masking policies for different regulatory contexts, and enabling model versioning and fallback across LLM providers. In smart-city and community-service environments, such browser-level safeguards can complement formal AI governance and urban cybersecurity measures.

ACKNOWLEDGMENT

This work was partially co-funded by the "Brindisi Smart City Port" project (PAC Infrastrutture, CUP: J85F20000710002).

REFERENCES

[1] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu, "Harnessing the power of llms in practice: A survey on chatgpt and beyond," 2023. [Online]. Available: https://arxiv.org/abs/2304.13712

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2025. [Online]. Available: https://arxiv.org/abs/2303.18223

[3] R. B. et. al., "On the opportunities and risks of foundation models," 2022. [Online]. Available: https://arxiv.org/abs/2108.07258

[4] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2402.00888

[5] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (llm) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, vol. 4, no. 2, p. 100211, Jun. 2024. [Online]. Available: http://dx.doi.org/10.1016/j.hcc.2024.100211

[6] S. Wilson and A. Dawson, "OWASP Top 10 for Large Language Model Applications — LLM01:2025 Prompt Injection," https://owasp.org/www-project-top-10-for-large-language-model-applications/assets/PDF/OWASP-Top-10-for-LLMs-v2025.pdf, Nov. 2024, oWASP GenAI Security Project. Published November 18, 2024.

[7] V. Benjamin, E. Braca, I. Carter, H. Kanchwala, N. Khojasteh, C. Landow, Y. Luo, C. Ma, A. Magarelli, R. Mirin, A. Moyer, K. Simpson, A. Skawinski, and T. Heverin, "Systematically analyzing prompt injection vulnerabilities in diverse llm architectures," *arXiv preprint arXiv:2410.23308*, 2024.

[8] C. Pathade, "Red teaming the mind of the machine: A systematic evaluation of prompt injection and jailbreak vulnerabilities in llms," *arXiv preprint arXiv:2505.04806*, 2025.