International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

Privacy and Security Issues in Generative AI: GenAI Shield, a Prompt-Level Privacy and Security Analyzer

Adrat Tejal Sakharam[1]; Barbuddhe Vedanti Sudhakarrao[2]; Awale Sakshi Ravindra[3]; Dr. Madhavi Shamkuwar[4]

[1-4]Department of MCA, Zeal Institute of Business Administration, Computer Application and Research (ZIBACAR), Narhe, Pune, India

Email: pteju6112@gmail.com **Orcid**: https://orcid.org/0009-0003-6417-6961

Email: barbuddhevedanti31@gmail.com **Orcid**: https://orcid.org/0009-0008-7621-241x

Email: mrawale217@gmail.com **Orcid**: https://orcid.org/0009-0002-6907-6145

Email: madhavi.Shamkuwar@zealeducation.com **Orcid**: https://orcid.org/0000-0002-9277-0912

Abstract

Users of Generative AI (GenAI) systems routinely paste personal data, passwords, API keys and internal documents into prompts, while the systems themselves face attacks such as prompt injection, and users have few tools that check a prompt before it is submitted. This paper presents the design and evaluation framework of GenAI Shield, a prompt-level privacy and security analyzer that combines pattern-based detection of personal data and credentials, rule-based detection of confidential markers, prompt-injection and suspicious content, a transparent project-specific risk score (0–100, four levels), and a rule-driven recommendation engine. The system (Python/Flask, SQLite, Chart.js) is under development; no module has yet been implemented, deployed or tested, and this paper reports the proposed design only. A synthetic Stage 1 dataset of 200 prompts in six categories was used to demonstrate the evaluation method (synthetic data was used for Stage 1 because the prototype is incomplete, and will be replaced by measured results in Stage 2); the detector outcomes in it are assumed and illustrative. Under that illustration, precision is 0.95, recall 0.79 and F1 0.86 against synthetic labels, with structured categories (personal data, credentials) flagged at 94.55% and semantic categories (confidential, injection, suspicious content) at 68.24%. The contribution is an architectural and methodological integration of privacy detection, threat detection, scoring and advice for GenAI prompts, reported with an explicit account of what is proposed versus what is measured.

Keywords

Generative AI; Privacy; Security; Prompt Injection; PII Detection; Risk Scoring

International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

1. Introduction

GenAI tools are now part of everyday study, coding and office work, but every prompt is a data transfer to a remote model. Language models can memorise and reveal training data, including personal details (Carlini et al., 2021; Lukas et al., 2023), and users may disclose information that should stay private, since whether a disclosure is appropriate depends on context (Mireshghallah et al., 2024). On the security side, prompt injection hides adversarial instructions in inputs or retrieved content (Greshake et al., 2023), and jailbreaks bypass safety training (Wei et al., 2023). Most defences work inside the model or the application. This research proposes GenAI Shield, a user-side analyzer that screens a prompt before submission, explains the risk and advises remediation, and it reports honestly which parts are proposed, which are illustrated on synthetic data, and which remain to be measured.

2. Literature Review

Surveys organise the field (Yao et al., 2024) and standards call for risk management of GenAI (National Institute of Standards and Technology [NIST], 2024; OWASP Foundation, 2025). Training-data extraction work shows that models leak memorised content, including personal data (Carlini et al., 2021; Lukas et al., 2023), but it concerns provider-held data, not what users type. Prompt-injection research moved from direct attacks to indirect ones that arrive through retrieved content (Greshake et al., 2023), and jailbreaks are explained by competing training objectives and mismatched generalisation (Wei et al., 2023). Secret scanning of public repositories combines pattern rules with entropy filtering (Meli et al., 2019), and text-anonymisation research warns that removing direct identifiers is insufficient (Lison et al., 2021). Guardrail tools such as Llama Guard (Inan et al., 2023) screen prompts for content safety, but are developer-oriented and do not produce a user-facing privacy and credential score. Table 1 summarises the key studies and their relevance.

Table 1. Literature Comparison

Study	Focus	Key Finding	Relevance to This Study
Carlini et al. (2021)	Training-data extraction	Verbatim training sequences, some personal, can be recovered from an LLM	Evidence of PII leakage risk
Lukas et al. (2023)	PII leakage	Scrubbing reduces but does not remove leakage	Motivates layered, not single-filter, checks
Mireshghallah et al. (2024)	Contextual privacy	Models disclose private information inconsistently across contexts	Shows limits of format-based detection
Greshake et al. (2023)	Indirect prompt injection	Retrieved content can carry hidden instructions	Basis for injection-pattern module
Wei et al. (2023)	Jailbreak failure modes	Competing objectives and mismatched generalisation	Explains why static filters can be bypassed
Meli et al. (2019)	Secret leakage	Exposed secrets are common; patterns plus entropy detect them	Precedent for credential module

International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

Study	Focus	Key Finding	Relevance to This Study
Lison et al. (2021)	Text anonymisation	Direct-identifier removal is insufficient	Residual-risk caution for PII module
Inan et al. (2023)	LLM guardrail	Input and output screening with a safety taxonomy is feasible	Closest related tool; different focus
NIST (2024); OWASP Foundation (2025)	Guidance	Risk-management profile; ranked LLM application risks	Frames risk families and scope

3. Research Gap, Problem Statement and Objectives

The gap is not that no relevant component exists, but that privacy detection, credential detection and injection screening are rarely combined, for short interactive prompts, into one user-facing workflow with an openly documented score and plain-language advice. Users must therefore judge prompt risk unaided. This research addresses that gap by proposing such an integrated analyzer. Whether it helps users is an empirical question left to Stage 2.

3.1 Research Questions

1. What are the major privacy and security risks associated with GenAI usage?

2. Which types of sensitive information are most relevant to prompt-level privacy detection?

3. How can prompt injection and other suspicious prompt patterns be identified?

4. How can these risks be represented by a transparent risk-scoring mechanism?

5. How can an integrated analysis framework help users make safer GenAI interactions?

6. What limitations remain when synthetic datasets are used for preliminary evaluation?

3.2 Research Objectives

1. To study privacy and security issues associated with GenAI systems.

2. To identify common risks: PII exposure, credential exposure, data leakage, prompt injection and suspicious inputs.

3. To review existing protection approaches for GenAI applications.

4. To design a conceptual framework for analysing prompt-level risks.

5. To propose the GenAI Shield application.

International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

6. To develop a transparent risk-scoring and recommendation framework.

7. To conduct a preliminary controlled evaluation on synthetic data and to define the evaluation methodology for real data.

4. System, Architecture and Methodology

Pipeline: User → prompt input → Flask backend → prompt-analysis engine → privacy detection (PII, confidential markers) → security threat detection (credentials, injection, suspicious content) → risk-scoring engine → recommendation engine → result dashboard and analysis history. Proposed stack: Python 3 / Flask backend; HTML, CSS, JavaScript and Bootstrap frontend; Chart.js for charts; SQLite with SQLAlchemy for storage; Pandas for statistics. The overall flow is shown in Figure 1 and the module status is listed in Table 2.

Figure 1 (to be drawn). Proposed system flow: User → web interface → Flask backend → prompt-analysis engine → [privacy detection | security-threat detection] → risk-scoring engine → recommendation engine → result dashboard, with analysis history stored in SQLite and aggregated with Pandas for the statistics view.

Table 2. Module Status (no module is implemented-and-verified at the time of writing)

Module	Status
Analyze Prompt, dashboard, history, statistics	Proposed; planned for first prototype
Regex-based PII detection with checksum validators	Proposed; planned for first prototype
Credential detection (signatures, keywords, entropy)	Proposed; planned for first prototype
Confidential-marker, injection-pattern and suspicious-content rules	Proposed; planned for first prototype
Risk scoring and recommendation engine	Proposed; planned for first prototype
NER-assisted PII, NLP/ML injection classifier, LLM-based analysis	Future implementation (Stage 2 or later)
Detection of data poisoning and model manipulation	Research concept only; not in scope

No database row counts, test results or timings are reported, because nothing has been executed yet.

4.1 Detection, Risk Scoring and Recommendation Methodology

Detection (proposed). Regular expressions with checksum validators (Luhn for cards, Verhoeff for Aadhaar-format numbers) find formatted personal identifiers; keyword rules find password statements and confidentiality labels; signature rules find key, token, private-key and connection-string shapes; a Shannon-entropy heuristic ($H = -\Sigma\, p_k \log_2 p_k$) flags long random-looking tokens, as in secret scanning (Meli et al., 2019); a known-phrase library covers instruction override, system-prompt extraction, role manipulation and authority claims. These rules are expected to miss paraphrased, obfuscated or optimised attacks.

International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

Risk score (proposed, project-specific, not an industry standard). Each detection has an assumed confidence c. Factor intensity is $S = 1 - \Pi(1 - c)$. The score is $R = 100 \times [1 - \Pi(1 - w_i S_i)]$ with assumed severity weights w: credential 0.85, prompt injection 0.65, data leakage/confidential 0.65, privacy 0.60, suspicious content 0.55. Levels: 0–24 LOW, 25–49 MEDIUM, 50–74 HIGH, 75–100 CRITICAL. Example: an email and a phone number (c = 0.6 each) give S = 0.84 and R = 50.4 (HIGH). The weights are design judgements, not calibrated values.

Recommendations (proposed). A rule table maps findings to advice: mask personal information; remove passwords and revoke or rotate keys; avoid submitting confidential information; review and sanitise suspected injection; and "do not submit without remediation" for HIGH and CRITICAL. The analyzer only advises; it does not alter the user's prompt.

4.2 Privacy-by-Design and Rule Provenance

The analyzer is intended to run on the user's own machine or server and to store only a masked copy or a hash of each prompt, so that it does not itself become a leakage point. Every detection rule will carry an identifier, a short description and an assumed confidence, so each finding in a result can be traced to a rule. Rules that produce false alarms on educational or programming prompts will be marked and worded as "review" advice.

5. Data and Evaluation Methodology

Evaluation uses standard metrics: a prompt is positive if it carries a seeded risk factor and is flagged if scored MEDIUM or above. Precision = TP/(TP+FP); recall = TP/(TP+FN); F1 = 2TP/(2TP+FP+FN); accuracy = (TP+TN)/total; specificity = TN/(TN+FP). Measuring these on the real prototype, with real or public prompt data and independent labels, has not yet been done.

Data Note (Stage 1 Development). Because the prototype is incomplete, the 200-prompt dataset analysed in Section 6 is synthetic, generated for Stage 1 so that the evaluation pipeline could be built and demonstrated. It was built from templates filled with obviously fake values (reserved example domains, fictional-range phone numbers, vendor-documented placeholder keys, invented names and companies), with no real credentials or personal data. It does not represent real users or production traffic. In addition, the "detected / not detected" outcomes and risk scores in Section 6 are assumed, illustrative values set by the authors, not produced by running GenAI Shield. The figures therefore illustrate the analytical method and are not empirical findings. In Stage 2 they will be replaced by measured results.

6. Analysis

Tables 3–6 present the descriptive analysis of the 200-prompt synthetic Stage 1 dataset described in Section 5. Table 3 gives its composition, Table 4 the expected risk levels, Table 5 the illustrative detection rates by category, and Table 6 the metrics and average scores.

Table 3. Dataset Composition (Synthetic Stage 1 Data, n = 200)

Category	Prompts	%
Safe	60	30.0

International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

Category	Prompts	%
Privacy risk (PII)	30	15.0
Credential risk	25	12.5
Data leakage / confidential	25	12.5
Prompt injection	35	17.5
Malicious / suspicious	25	12.5

The 70% share of risky prompts is a design choice that gives each detector enough cases; real traffic would contain far more benign prompts.

Table 4. Expected Risk Level (ground-truth labels; Synthetic Stage 1 Data, n = 200)

Risk level	Prompts	%
LOW	60	30.0
MEDIUM	47	23.5
HIGH	62	31.0
CRITICAL	31	15.5

Credential prompts can reach CRITICAL alone; CRITICAL cases in other categories are compound prompts with two co-occurring factors.

Table 5. Illustrative Detection Performance by Category (assumed outcomes, Synthetic Stage 1 Data)

Threat category	Cases	Detected	Detection rate (%)	Mean score
Privacy risk (PII)	30	28	93.33	45.60
Credential risk	25	24	96.00	71.84
Data leakage / confidential	25	18	72.00	45.96
Prompt injection	35	24	68.57	42.80
Malicious / suspicious	25	16	64.00	38.80
All risky prompts	140	110	78.57	—
Safe prompts (correctly passed)	60	54	90.00	8.40

The overall mean score across all 200 prompts is 36.43 (7,285 / 200).

International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

Table 6. Illustrative Confusion Matrix and Metrics (against synthetic ground-truth labels)

Measure	Value
True positives / False negatives	110 / 30
False positives / True negatives	6 / 54
Precision	0.9483
Recall	0.7857
F1-score	0.8594
Accuracy	0.8200
Specificity	0.9000

The 6 assumed false positives were safe prompts that mention the word "password" in a programming question (3), quote an attack phrase in an educational question (2), or use an example-format email (1). The 30 assumed false negatives are 2 privacy, 1 credential, 7 confidential, 11 injection and 9 suspicious-content cases. These metrics reflect evaluation against synthetic ground-truth labels and should not be interpreted as production performance.

7. Evaluation of the Implemented System

The Stage 1 analysis in Section 6 illustrates the evaluation method; it does not measure the analyzer. Measuring detection performance requires the implemented prototype, a labelled prompt set with independent annotation, and real or public benchmark data, none of which exist yet. A prototype-trial user study of whether the advice changes user behaviour has likewise not been run. No numerical values are reported here, consistent with this paper's discipline of not estimating a result it has not measured.

8. Discussion and Limitations

In the Stage 1 illustration, structured categories were flagged more often (52 of 55, 94.55%) than semantic categories (58 of 85, 68.24%), and undetected cases (30) outnumbered false alarms (6) five to one. This pattern was built into the assumed outcomes, so it does not test the hypothesis that pattern-based detection works better on structured than on semantic risks; it shows how that hypothesis would be examined with measured data. The pattern is, however, what the literature leads one to expect: format-based rules cannot capture context-dependent privacy (Mireshghallah et al., 2024) or paraphrased attacks (Wei et al., 2023). Of the research questions in Section 3.1, RQ1–RQ4 are addressed at the level of literature review and design, RQ6 is addressed through the synthetic illustration, and RQ5 is not answered and is deferred to Stage 2.

Limitations: the application is in development and no module is implemented or tested; the Stage 1 dataset is synthetic and its detector outcomes are assumed; labels were produced by the authors and share the scoring rubric, so real validity is untested; rule-based detection yields false positives and can be evaded; the risk score and weights are project-specific and uncalibrated; no real-user study exists; poisoning and model manipulation are studied only conceptually; and GenAI threats change quickly, so rules need continuing maintenance.

International Journal of Science and Engineering Science Research
Volume 2 | Issue 4 | October-December 2026 | ISSN 3049-2793
Peer Reviewed (Referred Journal)
DOI: https://doi.org/10.63671/ijsesr.v2i4.182

9. Ethical Considerations

As Stage 1 uses synthetic data, no human participants were involved and no real credentials or personal data were used. In Stage 2, any user study will require informed consent, and collected prompts will be stored masked or hashed. No formal institutional ethics-board approval is claimed at this MCA-project stage. No automated plagiarism-detection tool (for example Turnitin or Drillbit) has yet been run on this manuscript, which the department's policy (maximum 10% for text-based research) requires before final submission.

10. Conclusion and Future Work

This paper presented the design of GenAI Shield, a prompt-level privacy and security analyzer that integrates pattern-based and rule-based detection, a transparent project-specific risk score and a recommendation engine, together with a synthetic Stage 1 walkthrough of how it will be evaluated. The walkthrough, whose outcomes are assumed, illustrates reasonable detection of structured risks and weaker detection of semantic ones. Completing and testing the prototype, measuring detection on real or public data with independent labels, calibrating the score weights, and running a prototype-trial user study are the immediate next steps.

Stage 2 Development (Future Scope). Synthetic data was used in Stage 1 because the prototype is incomplete. In Stage 2, the assumed outcomes will be replaced by results measured on the implemented system, the dataset will be extended with permitted real or public prompts, and the analysis in Section 6 will be repeated. Later extensions include NER-assisted and ML-based detection, multilingual prompts, a browser extension, API-gateway integration and explainable scoring.

References

1. Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., & Raffel, C. (2021). Extracting training data from large language models. In 30th USENIX Security Symposium (pp. 2633–2650). USENIX Association.
2. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (pp. 79–90). ACM.
3. Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., & Khabsa, M. (2023). Llama Guard: LLM-based input-output safeguard for human-AI conversations (arXiv:2312.06674). arXiv.
4. Lison, P., Pilán, I., Sánchez, D., Batet, M., & Øvrelid, L. (2021). Anonymisation models for text data: State of the art, challenges and future directions. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (pp. 4188–4203). Association for Computational Linguistics.
5. Lukas, N., Salem, A., Sim, R., Tople, S., Wutschitz, L., & Zanella-Béguelin, S. (2023). Analyzing leakage of personally identifiable information in language models. In 2023 IEEE Symposium on Security and Privacy (SP). IEEE.
6. Meli, M., McNiece, M. R., & Reaves, B. (2019). How bad can it git? Characterizing secret leakage in public GitHub repositories. In Proceedings of the Network and Distributed System Security Symposium (NDSS). Internet Society.
7. Mireshghallah, N., Kim, H., Zhou, X., Tsvetkov, Y., Sap, M., Shokri, R., & Choi, Y. (2024). Can LLMs keep a secret? Testing privacy implications of language models via contextual integrity theory. In The Twelfth International Conference on Learning Representations (ICLR 2024).

DOI: https://doi.org/10.63671/ijsesr.v2i4.182

8. National Institute of Standards and Technology. (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile (NIST AI 600-1). https://doi.org/10.6028/NIST.AI.600-1
9. OWASP Foundation. (2025). OWASP Top 10 for LLM applications 2025. https://genai.owasp.org/llm-top-10/
10. Wei, A., Haghtalab, N., & Steinhardt, J. (2023). Jailbroken: How does LLM safety training fail? Advances in Neural Information Processing Systems, 36.
11. Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly. High-Confidence Computing, 4(2), 100211.