DataSIR:A Benchmark Dataset for Sensitive Information Recognition

Fan Mo[1,2,3], **Bo Liu**[2*], **Yuan Fan**[1,2], **Kun Qin**[2], **Yizhou Zhao**[2], **Jinhe Zhou**[2], **Jia Sun**[2], **Jinfei Liu**[1,3,4], **Kui Ren**[1,3]
[1]Zhejiang University
[2]DBAPPSecurity Co., Ltd
[3]State Key Laboratory of Blockchain and Data Security, Zhejiang University
[4]Hangzhou High-Tech Zone (Binjiang) Blockchain and Data Security Research Institute
[1]{fan.mo, fanyuan, jinfeiliu, kuiren}@zju.edu.cn
[2]{bo.liu, kian.qin, yichou.zhao1, jinhe.zhou, jia.sun}@dbappsecurity.com.cn

A Technical Appendices and Supplementary Material

A.1 Comparison of Results for Gemini with Different Format Transformations

Gemini attained optimal performance metrics for sensitive category and format transformation scenarios tasks, surpassing all comparator models in maximum achievable performance. The focus was then placed on Gemini's ability to recognize and restore both original and transformed data. The experimental results are shown in Table 1. In the main text section Experiments, due to space constraints, only four key observations were analyzed, as follows:

i) The LRAcc and DRAcc of total format transformed data is less than original data, which indicates that it is more difficult to recognize and restore data after format transformed.

ii) Gemini's recognition of URL-encoded data is the best, as URL encoding only involves transforming Chinese characters and some symbols, making it relatively easy for large language models to restore the original data and significantly enhancing the recognition of sensitive categories.

iii) Gemini's recognition of data transformed into binary, octal, and hexadecimal formats is poor. These transformations only affect numbers, and only the IMEI and IMSI (purely numeric) sensitive categories support such transformations. Due to the lack of contextual information in the sample data, large language models may confuse these with personal identifiers, mobile numbers, and MEID. They are more likely to identify them as more severe leaks (e.g., personal identifiers and mobile numbers), resulting in LRAcc values below 20%.

iv) Additionally, Gemini can almost fully restore binary and octal-transformed data to their original form, but it cannot distinguish between hexadecimal-transformed data and hexadecimal MAC addresses, leading to a DRAcc of 0% for hexadecimal format transformations.

In addition, further key observations have been noted, as follows.

i) The reason why the overall dataset's DRAcc is higher than LRAcc: Some sensitive data categories containing numbers are easily confused, such as Personal ID, IMEI, IMSI, Passport, MAC, Driver's License, MEID, etc. Therefore, LRAcc is relatively low. Some encoding methods are easily restored, especially ASCII encoding, Unicode encoding, etc. Data encoded through these encoding methods are more regular and easily restored, therefore, DRAcc is relatively high. Although the data is easily restored, large language models cannot accurately recognize the label of restored data. This can be foreseen from the LRAcc for original data (around 70%), which indicates that not all restored data can be recognized.

*Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

ii) The LRAcc for numerical capitalization is less than 50%, specifically because it only processes numerical data types. As mentioned earlier, sensitive data categories containing numbers are easily confused, which leads to a decrease in LRAcc.

iii) The DRAcc for simplified-to-traditional Chinese is lower than LRAcc, because large language models sometimes consider that the encoded data has undergone simplified-to-traditional Chinese transformation, but sometimes they also consider that it has not undergone transformation, leading to a decrease in DRAcc for simplified-to-traditional Chinese, which is lower than LRAcc.

iv) The DRAcc for character decomposition is also lower than LRAcc. Although large language models can correctly recognize the content obtained after decomposition, if large language models are not explicitly reminded to associate the decomposed content, they will not merge the decomposed content into the whole to restore the data. For example, "功" is decomposed into "工" and "力". The large language model will consider that there are two characters, "工" and "力", here.

Table 1: Comparison of Results for Gemini with Different Format Transformations

Type	**LRAcc (%)**	**DRAcc (%)**
Binary	18.00	98.00
Octal	18.00	98.00
Hexadecimal	16.00	0.00
ASCII encoding	69.57	95.74
Unicode encoding	71.39	97.17
UTF-8 encoding	72.43	95.53
Base64 encoding	59.02	66.47
URL encoding	86.02	97.49
HTML entity encoding	70.64	94.78
Morse encoding	63.37	69.77
Braille encoding	52.71	46.51
Nested encoding	57.68	60.21
Acrostic poetry	71.85	76.30
Character decomposition	66.35	61.54
Text inversion	68.57	57.96
Martian text	61.25	58.27
Simplified to traditional Chinese	74.04	50.96
Numerical capitalization	47.86	78.35
Inserting special characters	66.02	68.71
Inserting Chinese characters	80.14	85.82
Inserting English letters/numbers	65.38	58.65
All Above Format Transformed Data	64.39	75.26
Original data	72.58	95.08

A.2 Empirical Validation of Format Transformations in DataSIR

In DataSIR, a comprehensive empirical validation was conducted to ensure the realism and credibility of all 21 format transformation types. These transformations were not artificially constructed but were instead derived through large-scale observation and synthesis of real-world evasion tactics documented across industrial threat intelligence reports and adversarial NLP research.

A.2.1 Sources of Transformation Design

The selection of transformation types was informed by two complementary evidence sources: (1) adversarial techniques reported in recent academic literature, and (2) industrial incident analyses describing real-world evasion and obfuscation practices. By integrating insights from both domains, each transformation was designed to represent practically observable adversarial behaviors rather than synthetic perturbations, thereby enhancing the dataset's ecological validity.

A.2.2 Overlap with Existing Adversarial Research

The correspondence between the transformations adopted in DataSIR and those identified in recent adversarial studies and industrial analyses is summarized in Table 2. A substantial degree of overlap can be observed, indicating that the transformation space represented in DataSIR has been well aligned with real-world adversarial practices.

Table 2: Overlap between adversarial research and format transformations in DataSIR

Source	Evasion Scenario	Overlap	Example Types
HackAPrompt'23 [3]	Obfuscation and encoding	7	Base64, Chinese conversion, Martian text
Elastic PowerShell Report	Dynamic script evasion	5	Text inversion, Numeral caps, Decomposition
ThreatDown Report	Code obfuscation	7	Binary, ASCII, Unicode, Nested encoding
StructuralSleight'23[1]	Character-level obfuscation	8	Base64, Hex, URL, HTML encoding
LLM Jailbreak Survey[2]	Natural language obfuscation	6	Acrostic, Martian text, Inversion

A.2.3 Empirical Mapping to Real-World Evidence

To further substantiate the authenticity of each transformation, a detailed empirical mapping was constructed linking every transformation type to its corresponding real-world evidence, as shown in Table 3 . Each mapping is grounded in documented adversarial incidents or recognized obfuscation methodologies across both cybersecurity and NLP domains.

Table 3: Empirical mapping between 21 format transformations and real-world adversarial scenarios

Format Transformation	Real-World Adversarial Scenario	Supporting Evidence
Binary	Deserialization attacks, IoT malware injection	Microsoft Security Guidelines; Akamai Threat Report
Octal	Data leakage prevention evasion	Springer: Multi-base encoding strategies
Hexadecimal	LLM prompt injection, filter evasion	0din.ai; promptfoo
ASCII encoding	Data exfiltration	LinkedIn Copilot study; SCWorld report
Unicode encoding	Homoglyph phishing	MITRE CAPEC-71; heise Security
UTF-8 encoding	Overlong encoding bypasses	USD Labs: CVE-2023-26302
Base64 encoding	Credential hiding, malware distribution	OPSWAT; VMRay
URL encoding	Injection evasion	Google Cloud; Huawei Cloud WAF
HTML entity encoding	XSS evasion	OWASP guidelines
Braille encoding	Covert communication, LLM red teaming	arXiv; CyberInfoBlog
Nested encoding	Multi-layer obfuscation	Unit42; OWASP
Acrostic poetry	Phishing evasion	MIT Computational Linguistics; AAAI
Character decomposition	Chinese steganography	ResearchGate study
Text inversion	Script obfuscation	Elastic Security report
Martian text	Homoglyph and content evasion	Hexatic; AAAI
Simplified↔Traditional Chinese	Keyword filtering bypass	ResearchGate
Numerical capitalization	Financial fraud	OWASP Top 10-2021
Inserting special characters	Text obfuscation	Elastic report
Inserting Chinese characters	Phishing text	OWASP injection model
Inserting English letters/numbers	AI validation bypass	Elastic: numeral ratio detection

The above evidence collectively demonstrates that the format transformations in DataSIR are empirically grounded in real-world adversarial behavior, rather than hypothetical or synthetic perturbations. By aligning dataset construction with verified industrial incidents and academic findings, DataSIR

achieves strong ecological validity and transferability for adversarial robustness evaluation. This evidence-based justification enhances the transparency and credibility of the dataset's design process.

A.3 Definition and Explanation of Metrics

A.3.1 Definition

LRAcc (Label Recognition Accuracy)

Label Recognition Accuracy (LRAcc) is used to evaluate the model's ability to recognize different categories of sensitive data under various format transformation conditions. Essentially, this metric reflects the accuracy of the model in label recognition tasks—that is, the proportion of sensitive information categories correctly identified by the model.

DRAcc (Data Restoration Accuracy)

Data Restoration Accuracy (DRAcc) measures the model's ability to restore the content of data under different format transformation conditions. In essence, it is also an accuracy-based metric that indicates the extent to which the model can successfully recover the original data from formatted or perturbed inputs.

Precision

Precision measures the accuracy of the model's predictions for a specific sensitive data type. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

In this study, since sensitive data include multiple categories (e.g., mobile numbers, IMEI, IP addresses, etc.), correctly predicted samples within each category are treated as positive instances, and incorrectly predicted samples as negative instances. Precision is then computed separately for each category.

Recall

Recall evaluates the model's coverage in identifying a particular type of sensitive data. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

In other words, recall describes the **completeness** of the model's recognition for a specific sensitive data type.

F1-score

F1-score provides a balanced measure between precision and recall. It is the harmonic mean of the two, defined as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

This metric captures the trade-off between precision and recall, offering a more comprehensive view of model performance.

A.3.2 Explanation of Metric Adjustments

To quantitatively evaluate the model's recognition performance across different types of sensitive information, we reformulated each category's prediction task as a binary classification problem—treating correctly predicted samples within a category as positive instances and incorrectly predicted samples as negative instances. Based on this formulation, classical evaluation metrics such as Precision, Recall, and F1-score were employed to measure model performance.

It is important to note that the calculation of LRAcc is mathematically identical to Recall, but differs in evaluation scope. While Recall measures, for a specific category, the proportion of correctly detected instances within that category, LRAcc aggregates this computation across all categories (e.g., via micro-averaging or global statistics). Therefore, LRAcc can be viewed as a global recognition accuracy measure derived from a recall-based calculation.

A.4 Dependencies and Model Configurations

A.4.1 NLP Libraries and Models

Table 4: NLP Libraries and Model Configurations

Package/Model Name	Version	Parameter Settings
HanLP	2.1.1	hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SDP_CON_ELECTRA_SMALL_ZH
spaCy	3.8.4	en_core_web_sm / zh_core_web_sm
NLTK	3.8.1	averaged_perceptron_tagger
Presidio	2.2.359	hanlp.pretrained.mtl.CLOSE_TOK_POS_NER_SRL_DEP_SDP_CON_ELECTRA_SMALL_ZH / en_core_web_sm

A.4.2 LLM API Configurations

Table 5: Large Language Model (LLM) API Configurations

Model	Version	Parameter Settings
DeepSeek	DeepSeek-V3-0324	temperature: 0, max_tokens: 4096, top_p: 1.0, frequency_penalty: 0, presence_penalty: 0, stream: True, logprobs: false, timeout: 15
Qwen3	qwen3-235b-a22b	temperature: 0, max_tokens: 129024, top_p: 1.0, top_k: 0, presence_penalty: 0.5, stream: True, timeout: 15, seed: 1234
GPT	gpt-4.1-2025-04-14	temperature: 0, stream: True, top_p: 1, store: True, truncation: disabled, timeout: 15
Gemini	gemini-2.5-flash-preview-04-17	temperature: 0, stream: True, top_p: 0.95, top_k: 64, candidateCount: 1, timeout: 15

References

[1] B. Li, H. Xing, C. Huang, J. Qian, H. Xiao, L. Feng, and C. Tian. Exploiting uncommon text-encoded structures for automated jailbreaks in llms. *arXiv preprint arXiv:2406.08754*, 2024.

[2] L. Nan, D. Yidong, J. Haoyu, N. Jiafei, and Y. Ping. Jailbreak attack for large language models: A survey. *Journal of Computer Research and Development*, 61(5):1156–1181, 2024.

[3] S. Schulhoff et al. Ignore this title and hackaprompt: exposing systemic vulnerabilities of llms through a global scale prompt hacking competition (2023). *arXiv preprint arXiv:2311.16119*, 2024.