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) Institute of Blockchain and Data Security
[1]{fan.mo, fanyuan, jinfeiliu, kuiren}@zju.edu.cn
[2]{bo.liu, kian.qin, yichou.zhao1, jinhe.zhou, jia.sun}@dbappsecurity.com.cn

Abstract

With the rapid development of artificial intelligence technologies, the demand for training data has surged, exacerbating risks of data leakage. Despite increasing incidents and costs associated with such leaks, data leakage prevention (DLP) technologies lag behind evolving evasion techniques that bypass existing sensitive information recognition (SIR) models. Current datasets lack comprehensive coverage of these adversarial transformations, limiting the evaluation of robust SIR systems. To address this gap, we introduce DataSIR, a benchmark dataset specifically designed to evaluate SIR models on sensitive data subjected to diverse format transformations. We curate 26 sensitive data categories based on multiple international regulations, and collect 131,890 original samples correspondingly. Through empirical analysis of real-world evasion tactics, we implement 21 format transformation methods, which are applied to the original samples, expanding the dataset to 1,647,501 samples to simulate adversarial scenarios. We evaluate DataSIR using four traditional NLP models and four large language models (LLMs). For LLMs, we design structured prompts with varying degrees of contextual hints to assess the impact of prior knowledge on recognition accuracy. These evaluations demonstrate that our dataset effectively differentiates the performance of various SIR algorithms. Combined with its rich category and format diversity, the dataset can serve as a benchmark for evaluating related models and help develop future more advanced SIR models. Our dataset and experimental code are publicly available at `https://www.kaggle.com/datasets/fanmo1/datasir` and `https://github.com/Fan-Mo-ZJU/DataSIR`.

1 Introduction

The advancement of global digitalization is accompanied by the rapid and continuous circulation of data, which faces numerous leakage risks. In particular, LLMs such as GPT[32] and DeepSeek[20], accelerate the release of data value, but at the same time, their open application ecosystems introduce more security risks. For example, LLMs may inadvertently expose sensitive information during the instruction response and knowledge distillation processes. According to IBM Security's annual "Cost of a Data Breach Report"[25] released in July 2024, the global average cost of a data breach in 2023 rose to $4.88 million, reaching a new high, an increase of nearly 10% from $4.45 million in 2023, the largest increase since 2020.

*Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

Because of this, more and more countries and regions have enacted laws and regulations to ensure data security. In 1996, the U.S. enacted HIPAA[11] to protect the security and confidentiality of health information. In 2002, the Sarbanes-Oxley Act (SOX[39]) was introduced to combat financial fraud and improve the accuracy and transparency of corporate financial reporting. In 2018, the EU implemented the GDPR to harmonize data protection laws and strengthen the privacy rights of individuals, particularly concerning sensitive personal data. In 2020, California passed the CCPA[43] to empower consumers with greater control over their personal data and increase corporate transparency in data handling. In 2021, China introduced the PIPL[33] to protect individuals' personal information rights, to regulate data processing activities, and to balance the data protection and utilization.

Data leakage[16] can be a data loss of original data. For example, if hackers obtain the database passwords, they can directly access the original data in the database. There also exists non-original data leakage. For example, various format transformations can be performed on the data, such as Unicode encoding, and then after leakage, reverse Unicode encoding can restore the original data. Current data loss prevention techniques mainly focus on defending against the first type of leakage, and very few studies have focused on the second type. However, in recent years, attackers have leveraged various tools including LLMs, to generate format-transformed sensitive data, posing a serious challenge to traditional data protection systems.

This dataset focuses on sensitive data recognition, especially the recognition of sensitive data after format transformations. Our contributions are summarized in the following three points.

- **Multilingual and Rich-Regulations Coverage.** To ensure consistency and broad applicability, 26 representative sensitive data categories were selected based on major international regulations (e.g., HIPAA, SOX, GDPR, CCPA, PIPL). Sensitive information types that were commonly defined or overlapping across these regulations were identified and consolidated into a unified category set. And examples were provided in both Chinese and English. Through this process, a multilingual, regulation-aligned dataset was constructed to support cross-regional sensitive data recognition.
- **Extensive Format Transformations.** For each sensitive category, 21 transformation types (e.g., binary, octal, Morse code, insertion of digits or English words) are applied, resulting in 1,647,501 samples, which significantly enrich the diversity of sensitive data.
- **High-Quality Benchmark Dataset.** The dataset's quality was validated using various NLP and LLM methods and models, demonstrating strong differentiation capabilities across different categories and formats. It can serve as a robust benchmark for evaluating and developing future sensitive information recognition models.

The structure of this paper is as follows (see Figure 1), first explaining the dataset, then sampling the samples, respectively performing NLP model and LLM experiments, conducting in-depth discussions on the experimental results, and analyzing existing problems and optimization directions.

2 Related Work

Sensitive information detection technology is a core support for data privacy compliance [30, 11, 33, 43], however, its advancement has long been hindered by limitations in existing datasets, such as single language, single privacy regulation, single format, and lack of benchmark validations for recognition capability.

Existing datasets are mostly limited to a single standard: SPEDAC[4] constructs an English text classification benchmark based on GDPR; HealthDeID[7] focuses on HIPAA medical records; In addition, there is a lack of public datasets for PIPL, with only some news or encyclopedic corpora containing labeled data for names, addresses, and organizations. The faker package can generate data covering various sensitive information tags, but it does not follow any standard, and some data (e.g., Chinese ID numbers) do not conform to real-world validation logic[34, 5, 10, 29, 46].

In terms of format coverage, existing benchmarks are relatively limited, too. For example, DarkBERT[15] focuses on dark web text and industry jargon; TextBugger[18] involves format transformations limited to character manipulation and replacement, such as deletion, insertion, and substitution. Other common transformations, such as random text insertion, typos, word order replacement, and text rewriting, are primarily used in text augmentation during model training[9, 38, 24].

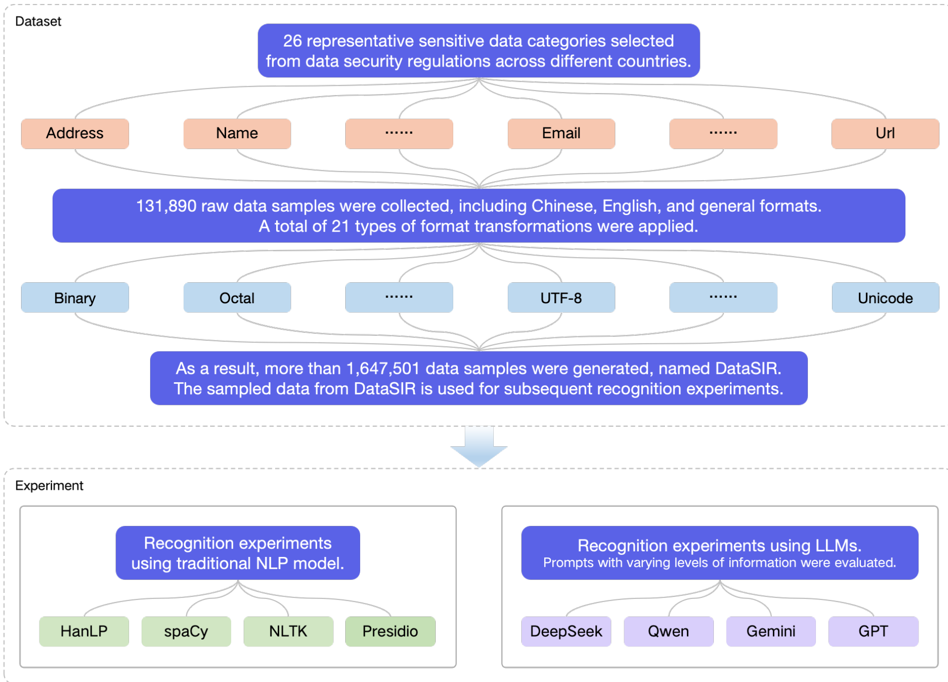

Figure 1: Flowchart of DataSIR.

Regarding the benchmark of recognition capabilities, very few studies or datasets have addressed this aspect. It is noteworthy that benchmarks such as PII-Bench[36] and PII-Scope[27] aim to "evaluate" the overall performance of privacy-preserving systems or "measure" privacy leakage risks, but they are not designed as algorithms for "detecting" formatted text transformations.

In addition, in the area of adversarial prompts, dataset[13] focuses on jailbreaking attacks on LLMs; dataset[26] emphasizes character injection in jailbreaking; and dataset[40] focuses more on profanity, discrimination, ethics, crimes, and financial privacy.

In summary, these datasets or techniques mostly cover only a single regulation, have limited format transformations, and pay insufficient attention to sensitive information (especially personal sensitive information). Because of these aforementioned issues, we introduce a novel dataset called DataSIR, which covers 26 common sensitive data categories from multiple information security regulations in Chinese and English, and contains 21 format transformations. Our experimental studies have proven that this dataset has enough benchmark differentiation for recognition capability of various methods. DataSIR can serve as a benchmark for sensitive information recognition models and also can act as a good reference for creating datasets for other related scenarios (e.g., adversarial prompt detection[22, 2]).

3 The DataSIR Dataset

3.1 Overall Introduction

First, we identified 26 representative sensitive data categories by extracting sensitive information types that appear in at least two major international regulations from countries and regions primarily using Chinese and English, ensuring broader relevance and cross-regulatory alignment. Then, we collected 131,890 original data samples from public sources. In addition to globally applicable sensitive categories, other sensitive categories include both Chinese and English examples, and also include samples with national differences from multiple countries. Finally, we constructed 21 common format transformations, selecting appropriate transformations for each category, resulting in 1,647,501 sensitive information samples, including the original data. Our dataset is publicly available at `https://www.kaggle.com/datasets/fanmo1/datasir`.

The representativeness of the dataset is mainly reflected in three aspects: **i)** Representativeness of sensitive categories: HIPAA, SOX, GDPR, CCPA, and PIPL cover major global economies such as the US, EU, and China. The sensitive information categories mentioned in these regulations have good representativeness. We collected hundreds of sensitive categories and statistically selected 26 representative ones. **ii)** Representativeness of example languages: In addition to globally applicable categories, other sensitive categories include both Chinese and English examples. Chinese and English are the two most widely used languages in the world, covering about 50% of the population, and also include samples with national differences from multiple countries. **iii)** Representativeness of format transformations: Format transformations cover English, Chinese, and universal examples (e.g., numbers, symbols), with around 10 matching transformations for each category, ensuring sample diversity.

Samples with national differences refer to the fact that the same sensitive category may have significant differences across countries. For example, mobile numbers in China are structured as: country code (+86) + carrier prefix (first 3 digits) + user number (last 8 digits). In the US, they are structured as: country code (+1) + area code (e.g., 917 for New York) + 7-digit local number. Mobile numbers in EU countries also vary. These national differences are considered in the collection of original data samples for mobile numbers. Additionally, formatting conventions such as spacing between digits, use of hyphens, or no separators are also considered. International dialing codes (+86) may include or exclude parentheses and the '+' sign. Similar considerations apply to other sensitive categories. An overview of the dataset is shown in Table 1. Some sensitive data categories have fewer instances because they are enumeration values. For example, marital status only has four possible values: single, married, divorced, and widowed.

Table 1: Overviews of DataSIR

Category	Covered Regulations	Language Involved	Original Count	Transformed Count	Total Count
Address	GDPR, PIPL, CCPA	Chinese/English	6000	72000	78000
Marital Status	GDPR, PIPL, CCPA	Chinese/English	8	104	112
Medical History	HIPAA, PIPL, CCPA	Chinese/English	6000	74838	80838
Name	GDPR, PIPL, CCPA	Chinese/English	6000	77607	83607
Nationality	GDPR, PIPL, CCPA	Chinese/English	482	6204	6686
Occupation	GDPR, PIPL, CCPA	Chinese/English	600	7542	8142
Organization	HIPAA, SOX, GDPR	Chinese/English	6000	73345	79345
Party	GDPR, PIPL, CCPA	Chinese/English	600	7402	8002
Religion	GDPR, PIPL, CCPA	Chinese/English	200	2569	2769
Date/Time	HIPAA, SOX	General	6000	48000	54000
Driver's License	GDPR, PIPL, CCPA	General	6000	66000	72000
Email	GDPR, PIPL, CCPA	General	6000	66000	72000
Personal ID	GDPR, PIPL, CCPA	General	6000	66000	72000
IMEI	GDPR, PIPL, CCPA	General	6000	84000	90000
IMSI	GDPR, PIPL, CCPA	General	6000	84000	90000
IPv4	GDPR, PIPL, CCPA	General	6000	66000	72000
IPv6	GDPR, PIPL, CCPA	General	6000	72000	78000
JDBC Connection String	GDPR, PIPL, CCPA	General	6000	66000	72000
Landline Number	HIPAA, CCPA	General	8000	96000	104000
MAC	GDPR, PIPL, CCPA	General	6000	72000	78000
MEID	GDPR, PIPL, CCPA	General	6000	66000	72000
Mobile Number	GDPR, PIPL, CCPA	General	8000	96000	104000
Passport	GDPR, PIPL, CCPA	General	6000	66000	72000
Postcode	GDPR, PIPL, CCPA	General	6000	66000	72000
Transaction Amount	GDPR, PIPL, CCPA, SOX	General	6000	48000	54000
URL	GDPR, PIPL, CCPA	General	6000	66000	72000

3.2 Detailed Description of Format Transformations

The following describes each format transformation type. We assigned letters A to U to each format transformation type for easy reference in subsequent discussions, and detailed explanations of these types will be provided later in the paper. The "Acrostic poetry" transformation was generated using an LLM, while the other 20 transformations were implemented through Python code to automatically generate the transformed data, which is accessible in the code repository at `https://github.com/Fan-Mo-ZJU/DataSIR`. "A. Binary": format transformation type "Binary" is represented by the letter "A". Then describes its transformation logic, and examples of the transformed data.

All 21 format transformations in DataSIR are based on real-world evasion scenarios, with their design informed by both empirical analyses and cutting-edge adversarial research ([35, 17, 28]), ensuring practical relevance and comprehensiveness. Typical examples include Base64 encoding, widely used in malware distribution, and Unicode encoding, commonly observed in phishing attacks; other transformations, such as hexadecimal and nested encoding, are also supported by empirical evidence. Collectively, these transformations comprehensively cover major security threat domains, ranging from web security and malware distribution to LLM jailbreaks, with detailed empirical references provided in the appendix.

- **A. Binary:** base-2 system using only 0 and 1 to represent data, where each binary digit corresponds to the smallest storage unit in computers.
  (e.g., 616655990822147 → 0110 0001 0110 0110 0101 0101 1001 1001 0000 1000 0010 0010 0001 0100 0111).
- **B. Octal:** base-8 system using digits 0-7.
  (e.g., 616655990822147 → 6 1 6 6 5 5 11 11 0 10 2 2 1 4 7).
- **C. Hexadecimal:** base-16 system using 0-9 and A-F.
  (e.g., 616655990822147 → 230d869481503).
- **D. ASCII encoding:** 4-bit encoding (0x00-0x7F) covering English letters, numbers and control characters, displayed in hexadecimal format.
  (e.g., China → 0x43 0x68 0x69 0x6E 0x61).
- **E. Unicode encoding:** universal character set assigning unique code points to characters worldwide.
  (e.g., China → `\u0043\u0068\u0069\u006e\u0061`).
- **F. UTF-8 encoding:** variable-length character encoding scheme representing any character in the Unicode standard.
  (e.g., China → `\x43\x68\x69\x6E\x61`).
- **G. Base64 encoding:** converting byte streams obtained through UTF-8 encoding into 64 printable characters (A-Z, a-z, 0-9, +, /)
  (e.g., China → `Q2hpbmE=`).
- **H. URL encoding:** special characters replaced by % followed by two hexadecimal values. For instance, space → %20, Chinese characters → UTF-8 encoded then converted).
  (e.g., 中国 → `%E4%B8%AD%E5%9B%BD`).
- **I. HTML entity encoding:** text first undergoes UTF-8 encoding then converts to HTML entities, represented by &entity_name; or &#entity_number; for preserved characters.
  (e.g., China → `China→China`).
- **J. Morse encoding:** using combinations of short (·) and long (--) signals to represent letters/numbers, separated by spaces between words.
  (e.g., China → `-.-.  ....  ..  -.  .-`).
- **K. Braille encoding:** text system representing characters through different arrangements of raised dot patterns.
- **L. Nested encoding:** applying different encodings multiple times to the same data (e.g., Base64 → UTF-8).
  (e.g., China → `\u0051\u0032\u0068\u0070\u0062\u006d\u0045\u003d`).
- **M. Acrostic poetry:** hiding information in the first character or initial letter of each sentence in text.
  (e.g., China → Crimson dragons dance through dynasties' dust, History hums in the Great Wall's crust, Ink-stained silk roads stitch heaven to earth, Nine bends of the Yellow River birth, A phoenix aflame—the East's rebirth.).
- **N. Character decomposition:** decomposing Chinese characters into components or strokes.
  (e.g., 功 = 工 + 力).
- **O. Text inversion:** reversing character sequence.
  (e.g., hello → olleh).
- **P. Martian text:** replacing original characters with visually similar ones.
  (e.g., 你 → ｨ尔) .
- **Q. Simplified to traditional Chinese:** one-to-one mapping of simplified Chinese characters to traditional ones.
  (e.g., 说 → 說).
- **R. Numerical capitalization:** converting numbers to Chinese characters.
  (e.g., 1 → 壹).
- **S. Inserting special characters:** inserting irrelevant symbols in the original text.
  (e.g., zero-width characters, emojis, special symbols #).

- **T. Inserting Chinese characters:** randomly inserting Chinese characters in text. (e.g., zero → zero 买它).
- **U. Inserting English letters/numbers:** inserting letters or numbers in text. (e.g., 你好 → 你OMG好).

It is particularly important to note that not all sensitive categories can undergo all 21 format transformations. Some transformations are applicable to Chinese, others to English, some to numbers, and others to symbols. Each category is applicable to around 10 transformations, with a minimum of 8 and a maximum of 14. Detailed information can be found in Table 2.

Table 2: Sensitive Category - Format Transformation Cross-Reference Table

Category	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**
Address	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	×	✓	✓	✓
Marital Status	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Medical History	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Name	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Nationality	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Occupation	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Organization	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Party	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Religion	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓
Date/Time	×	×	×	×	✓	✓	✓	×	✓	×	×	✓	×	×	×	✓	×	✓	✓	×	×
Driver’s License	×	×	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
Email	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	×	✓	×	×
Personal ID	×	×	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
IMEI	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
IMSI	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
IPv4	×	×	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
IPv6	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
JDBC Connection string	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	×	✓	×	×
Landline Number	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
MAC	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
MEID	×	×	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
Mobile Number	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
Passport	×	×	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
Postcode	×	×	×	✓	✓	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	×	✓	✓	×	×
Transaction Amount	×	×	×	×	✓	✓	✓	×	✓	×	×	✓	×	×	×	✓	×	✓	✓	×	×
URL	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	✓	×	×	✓	×	×

4 Experiments

4.1 General Experimental Preparation and Explanation

In each round of experiments, we sampled the dataset by randomly selecting 5 original data samples per sensitive category and performed recognition experiments on all format-transformed data associated with these 5 samples. We conducted 5 rounds of experiments, and reported the final label recognition accuracy (LRAcc, see Eq. (1)) as the average across all rounds.

For LLMs, we also separately calculated the data restoration accuracy (DRAcc, see Eq. (2), performing 5 rounds of experiments and taking the average as the final result. Since NLP models lack data restoration capabilities, this metric is not reported for them. "Restored Format-Transformed Data Matches Original" means that the two data items are identical after unifying case and removing whitespace characters.

$$\text{LRAcc} = \frac{N_C}{N_T} \tag{1}$$

$$\text{DRAcc} = \frac{N_M}{N_T} \tag{2}$$

N_C: Number of Samples with Fully Correct Sensitive Label Recognition.
N_M: Number of Samples Where Restored Format-Transformed Data Matches Original.
N_T: Total Number of Samples.

Metric Definitions and Explanations: To quantitatively assess recognition performance across categories, each prediction task is reformulated as a binary classification problem: correctly predicted samples are treated as positive instances, and incorrect ones as negative. Classical metrics—Precision, Recall, and F1-score—are then applied.

LRAcc is mathematically equivalent to Recall but differs in scope. While Recall evaluates detection within a single category, LRAcc aggregates this computation globally, serving as a recall-based overall accuracy indicator.

4.2 Comparative Experiments with NLP models

In our experiments, we evaluate models' end-to-end detection capabilities for diverse text transformations in a black-box setting, requiring no prior knowledge or preprocessing and allowing direct handling of multilingual, multi-format data. Our review indicates that existing specialized algorithms differ from the focus of this study in terms of research objectives ([21, 19]), experimental setups ([37, 42, 3]), and applicability ([45, 14]).

To ensure both generality and comparability, we adopt four mainstream NLP tools—HanLP, spaCy, NLTK, and Presidio—as baselines, chosen for their popularity on GitHub and their robust model-based capabilities in sensitive information processing.

HanLP[8] is a multilingual NLP toolkit for production environments, built-in with BiLSTM/CRF and pre-trained BERT, suitable for industrial scenarios such as Chinese word segmentation and entity recognition.

spaCy[12] is a high-performance production-grade NLP library with pre-trained model support, suitable for real-time text processing and multilingual scenarios.

NLTK[23] includes the WordNet dictionary, Brown corpus, and classic algorithms, suitable for academic research and small-scale data experiments.

Presidio[31] is an open-source privacy protection framework from Microsoft that combines rule engines and models to identify sensitive information (e.g., ID numbers, phone numbers), specifically for data compliance and de-identification.

We focused on the recognition capabilities of these NLP tools for data after format transformations in the 26 sensitive categories, directly using their built-in functionalities. The experimental results are shown in Table 3. All experiments were performed on a system with an Intel Core i7-1360P CPU and 32 GB of memory.

Table 3: Comparison of LRAcc for NLP Model Based Tools

Tool	Labels Count	List of Recognizable Labels	Original	Transformed	Overall
HanLP	8	Landline, Mobile Number, Date/Time, Postal Code, Amount, Address, Name, Organization	13.71%	**4.15%**	4.91%
spaCy	8	Date/Time, Amount, Nationality, Address, Name, Party Affiliation, Organization, Religious	13.29%	2.40%	2.98%
NLTK	3	Address, Organization, Name	2.59%	0.39%	0.56%
Presidio	12	IPv4, URL, Landline, Mobile Number, Date/Time, Email, Nationality, Address, Name, Party	**23.71%**	3.31%	**4.93%**

NLTK had the poorest recognition performance, identifying only 3 sensitive categories, with an overall label recognition accuracy of less than 1%. Presidio performed the best, identifying 12 sensitive categories, but its overall LRAcc was still less than 5%, as it integrates the spaCy model, so all its LRAcc are higher than those of spaCy. HanLP had the highest LRAcc for format-transformed data.

Since none of the four NLP models have data restoration capabilities, the DRAcc metric is not included. The overall LRAcc is also less than 5%, indicating that they have almost no recognition capability for multi-format-transformed data. Their recognition capability for the 26 original sensitive data categories was also weak, less than 25%. This suggests that the commonly used NLP models and tools in traditional data security solutions perform poorly in defending against advanced data leaks.

4.3 Comparative Experiments with Large Language Models

For LLMs, we selected four models: DeepSeek (deepseek-v3-0324), Qwen (qwen3-235b-a22b non-reasoning mode[41]), Gemini (gemini-2.5-flash-preview-04-17 non-reasoning mode[6]), and GPT (gpt-4.1). Due to the nature of the sensitive information recognition task in this paper: **i)** it does not involve complex reasoning processes, **ii)** the text to be analyzed is relatively short (even the longest acrostic poetry is only hundreds of characters long), **iii)** sensitive information recognition tasks are typically large-scale and require quick processing, making them unsuitable for long or time-consuming reasoning processes, additionally, the reasoning processes of commercial LLMs (except DeepSeek's reasoning model) are not accessible, which poses an insurmountable challenge for analyzing experimental results.

Therefore, we ultimately selected the above commonly used non-reasoning models offered by cloud service providers for the experiments. To ensure the stability of all experiments, we set the temperature parameter to 0 and defined the output format as JSON. Further configuration details of the models are provided in our GitHub repository.

In LLM prompts, the clearer the instruction and the richer the information provided, the stronger the model's ability to handle the task[1, 44]. Therefore, in this section, we designed three types of prompts with different information contents to elicit the models' recognition capabilities to varying degrees, aiming to demonstrate the dataset's ability to differentiate recognition capabilities.

A brief overview of the three prompts is provided below, while the complete versions are accessible in the code repository at `https://github.com/Fan-Mo-ZJU/DataSIR` due to space constraints.

No sensitive categories, no format transformations: the prompt only involves the task of identifying sensitive data without any specific labels or format transformation information.

With sensitive categories, no format transformations: the prompt includes the names of the 26 sensitive categories but no format transformation information.

With sensitive categories, with format transformations: the prompt includes the 26 sensitive label information, the specific logic of format transformations, and transformation examples.

The experimental results are shown in Table 4. As the information content in the prompts increases, the LRAcc also increases. In the scenario with sensitive categories and format transformations, the best LRAcc exceeds 60%, indicating that LLMs significantly outperform NLP models (with LRAcc less than 5%). If traditional data security solutions can integrate LLMs, the effectiveness of defending against data leaks would improve significantly and has the potential for further enhancement.

Table 4: Comparison of LRAcc for LLMs with Different Prompts

Prompts	DeepSeek LRAcc	Qwen LRAcc	Gemini LRAcc	GPT LRAcc
no label info, no format info	4.18%	5.68%	4.46%	**6.65%**
with label info, no format info	47.90%	47.55%	53.91%	**55.79%**
with label info, with format info	54.37%	55.97%	**65.04%**	64.30%

Gemini achieved the best performance in the scenario with sensitive categories and format transformations, showing the highest upper limit. The focus was then placed on Gemini's ability to recognize and restore both original and transformed data. Experimental results are presented in Table 5.

As shown, the impact of different format transformations on Gemini's recognition varies. Key observations include: **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 LLMs 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, LLMs 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

Table 5: Comparison of Results for Gemini with Different Format Transformation

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

original form, but it cannot distinguish between hexadecimal-transformed data and hexadecimal MAC addresses, leading to a DRAcc of 0% for hexadecimal format transformations.

Table 6: Comparison of Results for Gemini with Different Sensitive Categories

Category	Precision (%)	Recall (%)	F1-score (%)	DRAcc (%)
Address	62.65	99.08	76.76	61.85
Marital Status	90.80	95.62	93.15	89.69
Medical History	99.57	69.85	82.11	62.99
Name	65.11	92.51	76.43	76.08
Nationality	95.15	56.48	70.89	80.69
Occupation	97.65	62.09	75.91	71.64
Organization	40.89	78.05	53.67	65.85
Party	87.93	30.63	45.43	76.88
Religion	99.07	30.72	46.90	65.80
Date/Time	96.90	93.59	95.22	84.62
Driver’s License	16.67	0.67	1.28	75.00
Email	91.46	96.66	93.98	63.21
Personal ID	25.79	65.67	37.03	74.67
IMEI	26.03	21.87	23.77	83.20
IMSI	83.33	6.67	12.35	86.93
IPv4	95.62	94.67	95.14	87.00
IPv6	98.95	87.38	92.81	69.54
JDBC Connection string	97.39	99.67	98.52	69.67
Landline Number	62.69	74.46	68.07	76.92
MAC	62.77	89.23	73.70	76.31
MEID	68.29	18.73	29.40	65.22
Mobile Number	27.47	80.31	40.94	78.77
Passport	0.00	0.00	0.00	80.67
Postcode	73.50	77.67	75.53	93.00
Transaction Amount	71.72	31.56	43.83	64.89
URL	95.81	99.00	97.38	73.33

As shown in the Table 6, Gemini exhibits distinct performance patterns across different sensitive data categories. Key observations include: **i)** The model achieves F1-score above 90% in categories such as URL, JDBC Connection String, IPv4 and IPv6 Address, Email, and Date/Time. This indicates that even after format transformations, these categories of sensitive information with stable or distinctive formatting patterns can still be effectively recognized. **ii)** In contrast, the model’s performance declines significantly for sensitive data categories that rely heavily on semantic understanding. Categories such as Personal ID, Passport, Driver’s License, and Transaction Amount generally yield

F1-score below 40%. This performance degradation is mainly due to the fact that recognizing these categories of sensitive data depends more on contextual semantics and discourse cues, making it difficult to accurately identify them based solely on local features. **iii)** From the perspective of precision–recall balance, the Religion category is detected in a highly conservative manner (precision 99.07%, recall 30.72%), which effectively reduces false positives but leads to substantial under-detection. Conversely, the Mobile Number category exhibits the opposite trend (precision 27.47%, recall 80.31%), resulting in extensive over-matching. This contrast further reveals inconsistencies in the model's prediction strategies and highlights the lack of an effective confidence-balancing mechanism when dealing with complex or ambiguous cases.

5 Conclusion and Future Work

DataSIR focuses on sensitive data, especially data after format transformations. We first statistically analyzed sensitive categories mentioned in different countries' regulations and selected 26 representative categories. Then, we collected 131,890 original data samples and applied 21 format transformations, resulting in a dataset of 1,647,501 data samples. In experiments, we compared four NLP tools and four LLMs. The results revealed that NLP models commonly employed in traditional data security solutions exhibit poor performance in identifying data leaks, particularly when the data has undergone multiple format transformations, where recognition accuracy approaches zero. LLMs significantly outperform NLP models in recognition and have the capability to restore transformed data. The existing traditional data security solutions, if combined with LLM, would be significantly more effective in preventing data leaks. As the information content in LLM prompts increases, recognition accuracy improves, demonstrating that well-designed prompt engineering enhances recognition performance and that DataSIR has excellent differentiation in recognizing capabilities. We also discussed the difficulty of recognizing different format-transformed data and identified directions for future improvements.

The dataset also has some limitations: **i)** This paper only selected 26 representative sensitive data categories, and many other sensitive data categories were not included in the dataset. The types of format transformations can be increased, and the language coverage can also be expanded. **ii)** We only explored how increasing the information content of prompts can enhance the recognition capabilities of LLMs. **iii)** Samples in this dataset do not contain contextual information. For example, a name could belong to either a doctor or a patient. In this dataset, we cannot assign more specific category labels to these names due to differences in sensitivity: typically, a doctor's name can be publicly disclosed, whereas a patient's name cannot. We deeply understand the decisive role of context in real-world applications; however, building context-aware systems involves multi-dimensional scenario modeling and adversarial perturbations, which constitute a complex research challenge on their own. To ensure the rigor and focus of this study, we deliberately established a context-independent baseline for sensitive information recognition. This baseline provides a necessary comparative foundation for future, more complex context-aware research.

Future work includes the following: **i)** Continuously increasing the number of sensitive data categories, sample sizes, and format transformations to enhance the dataset's differentiation and better serve as a benchmark for sensitive information recognition models. **ii)** Exploring LLMs agents to further enhance recognition potential. **iii)** Integrating contextual signals (e.g., syntactic structure, semantic domains, and adversarial contexts) into future versions of DataSIR to better reflect real-world production scenarios and advance broader research in context-aware sensitive information recognition.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by the National Key RD Program of China (2022YFB3103401), NSFC (62472378, U23A20306), the Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars (LR25F020001), and the Zhejiang Province Pioneer Plan (2024C01074, 2025C01084).

References

[1] S. M. Bsharat, A. Myrzakhan, and Z. Shen. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4. *arXiv preprint arXiv:2312.16171*, 3, 2023.

[2] Cranot. Chatbot injections exploits. `https://github.com/Cranot/chatbot-injections-exploits`, 2024. Accessed: 2024-05-24.

[3] S. Eger, G. G. Şahin, A. Rücklé, J.-U. Lee, C. Schulz, M. Mesgar, K. Swarnkar, E. Simpson, and I. Gurevych. Text processing like humans do: Visually attacking and shielding nlp systems. *arXiv preprint arXiv:1903.11508*, 2019.

[4] G. Gambarelli, A. Gangemi, and R. Tripodi. Is your model sensitive? spedac: A new resource for the automatic classification of sensitive personal data. *IEEE Access*, 11:10864–10880, 2023.

[5] S. Garfinkel et al. *De-identification of Personal Information:*. US Department of Commerce, National Institute of Standards and Technology, 2015.

[6] Google Developers. Start building with Gemini 2.5 flash. `https://developers.googleblog.com/en/start-building-with-gemini-25-flash/`, April 2025. Online; accessed 2025-04-17.

[7] T. Hartman, M. D. Howell, J. Dean, S. Hoory, R. Slyper, I. Laish, O. Gilon, D. Vainstein, G. Corrado, K. Chou, et al. Customization scenarios for de-identification of clinical notes. *BMC medical informatics and decision making*, 20:1–9, 2020.

[8] H. He and J. D. Choi. The stem cell hypothesis: Dilemma behind multi-task learning with transformer encoders. *arXiv preprint arXiv:2109.06939*, 2021.

[9] J. He, L. Wang, J. Wang, Z. Liu, H. Na, Z. Wang, W. Wang, and Q. Chen. Guardians of discourse: Evaluating llms on multilingual offensive language detection. *arXiv preprint arXiv:2410.15623*, 2024.

[10] P. Henderson, M. Krass, L. Zheng, N. Guha, C. D. Manning, D. Jurafsky, and D. Ho. Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Advances in Neural Information Processing Systems*, 35:29217–29234, 2022.

[11] HHS. The hipaa privacy rule. `https://www.hhs.gov/hipaa/for-professionals/privacy/index.html`, 2024. Accessed: 2024-09-27.

[12] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, et al. spacy: Industrial-strength natural language processing in python. 2020.

[13] jackhhao. jailbreak-classification. https://huggingface.co/datasets/jackhhao/jailbreak-classification, 2023. Accessed: 2023-09-30.

[14] Z. Jiang, Z. Gao, G. He, Y. Kang, C. Sun, Q. Zhang, L. Si, and X. Liu. Detect camouflaged spam content via stoneskipping: Graph and text joint embedding for chinese character variation representation. *arXiv preprint arXiv:1908.11561*, 2019.

[15] Y. Jin, E. Jang, J. Cui, J.-W. Chung, Y. Lee, and S. Shin. Darkbert: A language model for the dark side of the internet. *arXiv preprint arXiv:2305.08596*, 2023.

[16] G. Kambala. Data privacy in cloud computing: A comparative study of privacy preserving techniques. *International Journal of Scientific Research and Management (IJSRM)*, 12(6), 2025.

[17] 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.

[18] J. Li, S. Ji, T. Du, B. Li, and T. Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

[19] L. Li and X. Qiu. Tavat: Tokenaware virtual adversarial training for language understanding. more: Understanding word-level textual adversarial attack via n-gram frequency descend. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 823–830, 2020.

[20] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[21] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*, 2020.

[22] Y. Liu, G. Deng, Y. Li, K. Wang, Z. Wang, X. Wang, T. Zhang, Y. Liu, H. Wang, Y. Zheng, et al. Prompt injection attack against llm-integrated applications. *arXiv preprint arXiv:2306.05499*, 2023.

[23] E. Loper and S. Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[24] D. Macko, R. Moro, A. Uchendu, I. Srba, J. S. Lucas, M. Yamashita, N. I. Tripto, D. Lee, J. Simko, and M. Bielikova. Authorship obfuscation in multilingual machine-generated text detection. *arXiv preprint arXiv:2401.07867*, 2024.

[25] Mindgard. Cost of a data breach report 2024. `https://www.ibm.com/reports/data-breach`, 2024. Accessed: 2024-04-25.

[26] Mindgard. evaded-prompt-injection-and-jailbreak-samples. `https://huggingface.co/datasets/Mindgard/evaded-prompt-injection-and-jailbreak-samples`, 2025. Accessed: 2025-04-25.

[27] K. K. Nakka, A. Frikha, R. Mendes, X. Jiang, and X. Zhou. Pii-scope: A comprehensive study on training data pii extraction attacks in llms. *arXiv preprint arXiv:2410.06704*, 2024.

[28] 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.

[29] J. Neerbek, M. Eskildsen, P. Dolog, and I. Assent. A real-world data resource of complex sensitive sentences based on documents from the monsanto trial. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1258–1267, 2020.

[30] Official Journal of the European Union. General data protection regulation. `https://gdpr-info.eu/`, 2018. Accessed: 2018-05-23.

[31] P. Ohm. Sensitive information. *S. Cal. L. Rev.*, 88:1125, 2014.

[32] OpenAI. Gpt-4.1. `https://openai.com/index/gpt-4-1/`, 2025. Accessed: 2025-04-15.

[33] PIPL. Personal information protection law of the people's republic of china. `https://www.gov.cn/xinwen/2021-08/20/content_5632486.htm`, 2021. Accessed: 2021-08-20.

[34] A. H. Razavi and K. Ghazinour. Personal health information detection in unstructured web documents. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pages 155–160. IEEE, 2013.

[35] 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.

[36] H. Shen, Z. Gu, H. Hong, and W. Han. Pii-bench: Evaluating query-aware privacy protection systems. *arXiv preprint arXiv:2502.18545*, 2025.

[37] W. Simoncini and G. Spanakis. Seqattack: On adversarial attacks for named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 308–318, 2021.

[38] S. Sood, J. Antin, and E. Churchill. Profanity use in online communities. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1481–1490, 2012.

[39] SOXCPA. The sarbanes-oxley act. `https://sarbanes-oxley-act.com/`, 2002. Accessed: 2002-07-30.

[40] H. Sun, Z. Zhang, J. Deng, J. Cheng, and M. Huang. Safety assessment of chinese large language models. *arXiv preprint arXiv:2304.10436*, 2023.

[41] Q. Team. Qwen3, April 2025.

[42] C. Udomcharoenchaikit, P. Boonkwan, and P. Vateekul. Towards improving the robustness of sequential labeling models against typographical adversarial examples using triplet loss. *Natural Language Engineering*, 29(2):287–315, 2023.

[43] `StateofCaliforniaDepartmentofJustice`. California consumer privacy act (ccpa). https://oag.ca.gov/privacy/ccpa, 2024. Accessed: 2024-05-13.

[44] L. Weng. Prompt engineering. *lilianweng.github.io*, Mar 2023.

[45] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant. Detecting homoglyph attacks with a siamese neural network. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 22–28. IEEE, 2018.

[46] K. Zhang and X. Jiang. Sensitive data detection with high-throughput machine learning models in electrical health records. In *AMIA Annual Symposium Proceedings*, volume 2023, page 814, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We include the main claims in the abstract and introducion.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss limitations in Section 5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: the paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our dataset and experimental code are publicly available. We provide all experimental details so that all readers can reproduce the experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The main dataset used in our experiments is publicly available on Kaggle with the specified link at `https://www.kaggle.com/datasets/fanmo1/datasir`. We also released our code repository at `https://github.com/Fan-Mo-ZJU/DataSIR`.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the original data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the implementation details in Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We did not conduct experiments with error bars. Running LLMs requires extensive time and compute, thereby making it generally impractical to runs all samples.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the description of compute resources in Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and conform to the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Negative societal impacts: Some readers may learn to leak data by using format transformation methods after reading this paper. Positive societal impacts: This article points out the fact that the existing traditional data security solutions, if combined with LLM, would be significantly more effective in preventing data leaks.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We have implemented safeguards by collecting each sensitive data individually, effectively minimizing the risk of deriving additional sensitive information through data association.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: To the best of our knowledge, we adequately cite and mention all used assets in this paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce new assets in Section 3

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.