You Have Been LaTeXpOsEd: A Systematic Analysis of Information Leakage in Preprint Archives Using Large Language Models

Richard A. Dubniczky[1,2], Bertalan Borsos[1], Norbert Tihanyi[1,3],
[1]Eötvös Loránd University, Budapest, Hungary
[2]ZEISS Digital Innovation, Dresden, Germany
[3]Technology Innovation Institute, Abu Dhabi, United Arab Emirates

***Abstract*—The widespread use of preprint repositories such as arXiv has accelerated the communication of scientific results but also introduced overlooked security risks. Beyond PDFs, these platforms provide unrestricted access to original source materials, including LaTeX sources, auxiliary code, figures, and embedded comments. In the absence of sanitization, submissions may disclose sensitive information that adversaries can harvest using open-source intelligence. In this work, we present the first large-scale security audit of preprint archives, analyzing more than 1.2 TB of source data from 100,000 arXiv submissions. We introduce LaTeXpOsEd, a four-stage framework that integrates pattern matching, logical filtering, traditional harvesting techniques, and large language models (LLMs) to uncover hidden disclosures within non-referenced files and LaTeX comments. To evaluate LLMs' secret-detection capabilities, we introduce LLMSec-DB, a benchmark on which we tested 25 state-of-the-art models. Our analysis uncovered thousands of PII leaks, GPS-tagged EXIF files, publicly available Google Drive and Dropbox folders, editable private SharePoint links, exposed GitHub and Google credentials, and cloud API keys. We also uncovered confidential author communications, internal disagreements, and conference submission credentials, exposing information that poses serious reputational risks to both researchers and institutions. We urge the research community and repository operators to take immediate action to close these hidden security gaps. To support open science, we release all scripts and methods from this study but withhold sensitive findings that could be misused, in line with ethical principles. The source code and related material are available at the project website: https://github.com/LaTeXpOsEd**

***Index Terms*—Artificial Intelligence, Large Language Models, Open-Source Intelligence (OSINT), Information Leakage**

I. Introduction

Scientific publishing has grown rapidly in recent decades, rising from roughly 1 million papers per year at the start of the 2000s to almost 2.9 million by 2020 [1]. In 2025, knowledge sharing is faster than ever, driven by growing pressure on researchers to rapidly communicate new discoveries. The traditional system of waiting months or even years for peer review can slow the impact of new discoveries and leave publications outdated, especially in fast-moving fields such as *artificial intelligence (AI)*, biotechnology, quantum computing, and genomics, just to name a few.

To address this challenge, preprint repositories such as arXiv have become foundational to modern scientific communication, enabling researchers to share their work as soon as results are ready, often well before formal peer-reviewed publication. The published preprint is freely available to everyone, supporting open science and providing a timestamped record with a DOI that verifies authorship and establishes who first introduced the idea.

Although this openness is highly valuable for advancing science, it also introduces new risks, as it enables *open-source intelligence (OSINT)* [2] activities targeting these documents. A key feature of platforms like arXiv is that they encourage authors to upload not only the final PDF but also the underlying source files used to prepare the manuscript, which can be valuable for reproducing the final document. Submissions often include LaTeX code, bibliographic files, figures, supplementary data, and even code. The standard submission process, favored for its speed and ease, typically lacks automated checks for accidental disclosures within supporting files. As a result, research artifacts that were once considered private, such as embedded comments, personal shared drive links, credentials to private repositories, metadata, and auxiliary files, can now be easily accessed by anyone who downloads the submission. Authors who focus primarily on scientific content may overlook the importance of sanitizing their submission files, potentially leading to the unintentional exposure of confidential or sensitive data.

While preprint servers vary in reputation—from highly trusted platforms to less moderated ones—all should be considered potential sources of data leakage. Table I summarizes source file availability across eight major preprint repositories.

TABLE I: Source availability from various preprint repositories.

Archive	Owner	Source	Entries
arxiv.org	Cornell University	●	> 2.8M
ssrn.com	Elsevier	○	> 1.7M
hal.science	CCSD	◐	> 1.5M
zenodo.org	CERN	◐	> 100k
preprints.org	MDPI	○	> 100k
osf.io	Center for Open Science	◐	> 25k
figshare.com	Digital Science	◐	> 5.4k
techrxiv.org	IEEE	●	> 1k

Legend: ● = Always available, ◐ = Optionally available, ○ = Not available.

Analyzing large volumes of source material using traditional tools such as email or credential harvesting [3] may uncover basic information, but these methods are often ineffective at identifying context-dependent disclosures embedded in LaTeX comments or internal author discussions that lack consistent patterns. These more in-depth insights require contextual understanding that older methods cannot provide. The emergence of *large language models (LLMs)*, which consistently achieve state-of-the-art performance across a wide range of *natural language processing (NLP)* tasks, makes them especially effective for this kind of analysis. Their ability to understand context, extract meaning, and interpret unstructured text enables them to uncover hidden or unintended disclosures in complex scientific documents.

A. Main Contribution

In this paper, we present the `LaTeXpOsEd` framework, a novel methodology that integrates entity extraction, pattern matching, and the language understanding capabilities of LLMs to automatically analyze hundreds of thousands of LaTeX source files together with auxiliary materials, encompassing 1.2 TB of data.

The main stages of the framework are shown in Figure 1 and are discussed in detail in the methodology section (Section III). To the best of our knowledge, this is the first systematic approach aimed at detecting sensitive data exposure in scientific preprints at this scale. Our study is driven by the following research questions:

1) **RQ1**: What kinds of sensitive information are exposed in preprint source files and comments, and how often do these disclosures occur?
2) **RQ2**: How effective are traditional pattern matching techniques and LLMs at detecting contextually hidden sensitive data within preprint source materials?

To answer these questions, we present four key contribution:

- **LaTeXpOsEd Framework**: We propose a new methodology that integrates traditional methods with advanced LLMs to systematically process LaTeX source files and accompanying materials from preprint archives, enabling the detection of sensitive information leaks.
- **Large-Scale Preprint Analysis**: We conduct the first comprehensive security audit of preprint source files, examining over 1.2 TB of data across 100 000 arXiv submissions. Comments and companion materials are systematically analyzed to uncover hidden information.
- **LLMSec-DB Benchmark**: We developed `LLMSec-DB`, a dataset designed to evaluate the ability of different LLMs to detect hidden secrets across diverse materials, including PDFs, text sources, and LaTeX files. Using this benchmark, we evaluated 25 state-of-the-art models.
- **Automated Data Analysis with LLMs**: Building on `LLMSec-DB`, we deployed the most cost-efficient open-source state-of-the-art LLMs to systematically extract sensitive data from preprints. This large-scale analysis uncovered thousands of instances of *ersonally identifiable information (PII)*, hundreds of credential leaks, private Google Drive access links, sensitive tokens, and shared folders unintentionally exposed in source files.

To support open science, we have made the `LLMSec-DB` dataset and all scripts used in this research publicly available on the project website: https://github.com/LaTeXpOsEd.

B. Ethical Considerations

During this research, we systematically analyzed over 100,000 preprint submissions comprising millions of associated files and identified a broad range of sensitive disclosures, including PII, leaked credentials, private repository links, and shared storage access tokens. To ensure ethical compliance, no raw sensitive data or PII was disclosed at any stage of the study, and all examples presented in this paper are anonymized or aggregated to prevent the re-identification of individuals or organizations. Our goal is to highlight broader risks and provide actionable recommendations, not to expose specific authors or institutions.

II. Related Literature

Research on arXiv as a data source has progressed from basic bibliometric analyses to sophisticated studies of scientific collaboration patterns [4]. The foundational work in arXiv analysis began with the unarXive dataset by Saier et al. [5], which represents one of the most significant contributions by providing over one million documents with 29.2 million citation contexts from arXiv publications. This also annotates in-text citations with global identifiers and links them to the Microsoft Academic Graph, creating a foundation for large-scale scholarly analysis such as citation recommendation, context analysis, and document summarization. It provides stable groundwork for the in-depth analysis of LaTeX source files. More recently, researchers have applied sophisticated analytical frameworks to arXiv data. A notable 2025 study by Hepler [6] examined 1,938 algorithmically discovered topics across 121,391 papers from arXiv metadata during 2020-2025, revealing structural patterns in mathematical research collaboration networks. This work showed that the popularity of research topics is closely linked to their network structure, with popular topics forming modular "schools of thought" and niche topics maintaining hierarchical core–periphery structures.

In a 2025 study, Pei et al. [4] examined 1.6 million LaTeX files to explore hidden patterns in scientific collaboration. The authors developed a novel methodology to examine author-specific macros in source code from papers authored by 2 million scientists between 1991 and 2023. Their analysis revealed that contribution patterns often contradicted traditional assumptions about authorship order. They also found a section-specific division of labor: some authors primarily contributed to conceptual sections such as the *introduction* and *discussion*, while others focused on technical sections like *methods* and *experiments*. Finally, they observed that collaboration norms

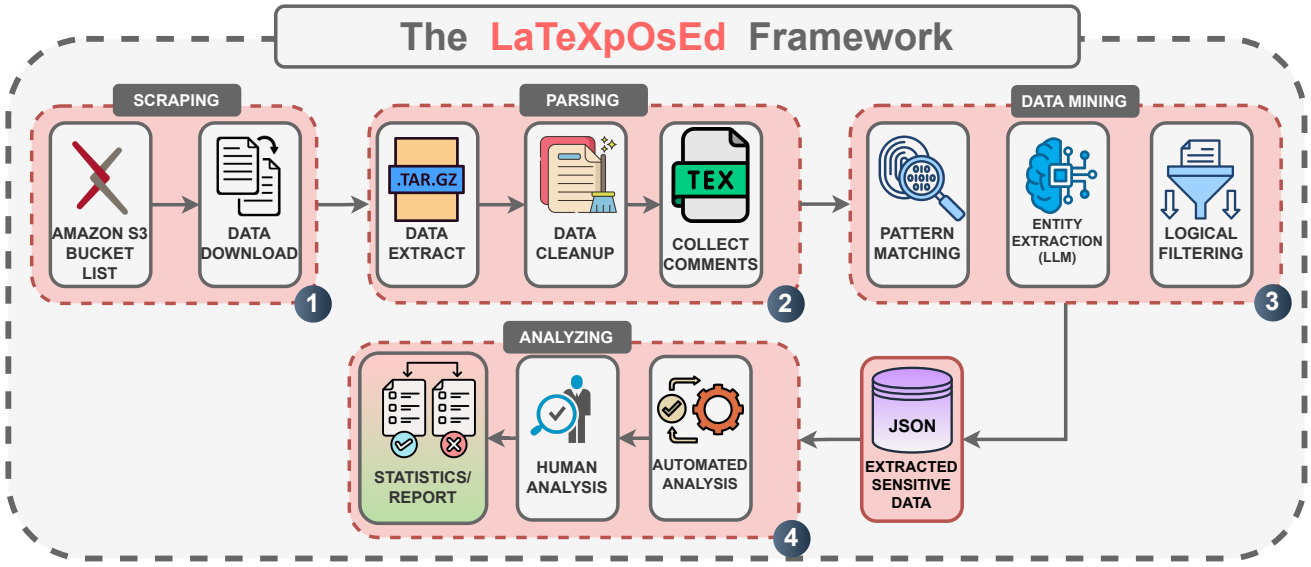

Fig. 1: The LaTeXpOsEd framework: a four-step process for scraping, parsing, mining, and analyzing documents.

varied across disciplines, with notable differences between fields such as computer science and economics.

Several studies have applied computational linguistics techniques to extract insights from arXiv content [7], [8]. A comprehensive analysis of quantitative finance papers from 1997 to 2022 [9] employed text mining techniques and NLP methods to analyze entire documents, including references, uncovering topic trends and identifying the most cited researchers and journals. The work demonstrated how NLP approaches can reveal patterns in scientific discourse that are invisible in traditional bibliometric analysis.

The APARA (AI Academic Research Aggregator) [10] system, developed by Anupama et al., uses the arXiv API combined with NLP to automatically extract metadata and summarize research contents, representing efforts to systematically process the growing volume of preprint literature.

In a comprehensive benchmark [11], Meuschke et al. evaluated ten freely available tools for extracting document metadata, bibliographic references, tables, and other content elements from academic PDF documents using a dataset of 500K pages from arXiv papers. This work established GROBID as the leading tool for metadata and reference extraction while revealing significant challenges in extracting complex elements like equations and tables.

The ability of LLMs to extract structural features and hidden information from raw text has already been demonstrated in other domains. In 2024, Tihanyi et al. [12] introduced a benchmark showing that LLMs can solve complex tasks and extract important structural features from raw text. Building on this, Dubniczky et al. [13] presented the CASTLE Benchmark in 2025, demonstrating that LLMs can also detect vulnerabilities and secrets in C source code.

In [14], Rahman et al. proposed a hybrid approach that combines regex-based candidate extraction with LLM-based classification to detect secrets such as API keys, tokens, and credentials in source code. They demonstrated that a fine-tuned LLaMA-3.1 8B model achieved strong performance (F1 ≈ 0.985), substantially reducing false positives compared to regex-only baselines.

In all cases, when LLMs were applied to extract meaningful information from large volumes of unstructured text, they achieved high accuracy and precision. In this study, we aim to leverage LLMs to extract sensitive data from a large collection of arXiv submissions. To the best of our knowledge, this is the first systematic study to apply advanced LLM-based techniques for extracting sensitive information from large-scale arXiv preprints.

III. Methodology

Our methodology is divided into four main stages, as illustrated in Figure 1. Scraping (collecting the data), Parsing (structuring and cleaning the source files), Data Mining (applying pattern matching and LLM-based extraction), and Analyzing (categorizing and interpreting the findings).

A. Stage 1 - Scraping

We initially attempted to acquire the papers through the `export.arxiv.org` service by issuing HTTP requests to the `/src/` endpoints. However, this approach proved inefficient and placed a significant load on the server infrastructure. Despite implementing batch requests, we frequently encountered rate-limiting and had to perform repeated retries. As a consequence, the effective throughput was reduced to approximately seven papers per minute, resulting in an estimated total of 240 hours to download 100,000 papers.

To improve efficiency, we accessed arXiv's Amazon S3 storage, which provides `tar.gz` archives containing about 125 papers each. An accompanying XML index maps individual papers to their respective archives, allowing us to identify

TABLE II: Summary of related work analyzing preprint archives and arXiv datasets. Our work is the first systematic study focusing on sensitive information extraction from LaTeX sources.

Year	Study	Contribution
2020	Saier et al.	Constructed large-scale dataset of 1M+ arXiv papers with citation contexts, linked to Microsoft Academic Graph.
2023	Meuschke et al.	Benchmarked 10 tools (e.g., GROBID) for metadata, bibliographic reference, and table extraction from PDFs.
2024	Bianchi et al.	Applied NLP/text mining to quantitative finance papers (1997–2022) to uncover trends and citation patterns.
2025	Hepler	Analyzed 121k arXiv papers (2020–2025) to map collaboration networks and structural patterns of research topics.
2025	Pei et al.	Extracted hidden author contributions from 1.6M LaTeX files, revealing division of labor and collaboration norms.
2025	Anupama et al.	Built AI-powered aggregator to summarize and index arXiv research using API and NLP.
2025	Rhman et al.	Proposed a hybrid method combining regex-based extraction with LLM classification to detect secrets (API keys, tokens, credentials) in source code.
2025	**Our Work**	First large-scale systematic security audit of 100k arXiv submissions (1.2 TB data). Extracted PII, credentials, tokens, and hidden author communications using LLMs and pattern matching.

808 archives needed to retrieve slightly over 100,000 papers. Overall, the complete download amounted to approximately 400 GB of compressed data (about 1.2 TB uncompressed), incurring a total transfer cost of $37.

B. Stage 2 - Parsing

In the parsing stage, we decompressed the `tar.gz` archives and extracted the relevant information into machine-readable JSON list files to facilitate downstream analysis. Approximately 8% of the downloaded submissions lacked source code and were available only as PDF files. These were excluded from further processing, leaving a set of roughly 92,000 papers with accessible source material. For each of these papers, we exported all textual content into a dedicated dump file to enable systematic pattern-matching analysis. We also extracted all LaTeX comments for LLM processing and mapped each paper's logical structure to detect auxiliary files not referenced in the final manuscript[1].

Additionally, to improve processing efficiency, we removed image files such as JPEGs and PNGs. However, before deletion, we performed an EXIF metadata check [15] to determine whether any sensitive information was embedded in the images. While most modern platforms automatically strip EXIF data for privacy protection, this process is not applied by the

[1]Here we note that we assume any content intentionally included in the final LaTeX paper—even credentials, if present—is considered part of the paper itself and not treated as secret or hidden information.

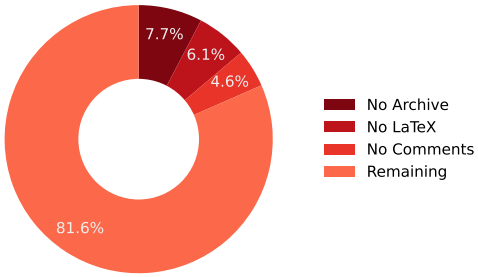

Fig. 2: Ratio of papers without usable comments in LaTeX source files.

arXiv servers. As a result, these files provided a valuable starting point for analysis, leading to the identification of nearly 1,200 images containing sensitive metadata. The types of data represented vary significantly. While device information (e.g., the camera used) or software details (such as the exact version of Photoshop) may already raise concerns, in over 600 cases the metadata contained GPS coordinates, potentially revealing the precise location where a photo was taken. In some instances, this could expose a researcher's home address (when tied to a profile picture) or the location of research facilities (when images capture experimental equipment). A more in-depth analysis is provided in Section IV.

Figure 2 illustrates that approximately 18% of the analyzed papers lack usable LaTeX comments for subsequent processing. This subset includes submissions with no source archives, archives devoid of LaTeX code, or LaTeX files that contain no comments. Notably, 4.6% of papers fall into the last category, where LaTeX code exists but all comments appear to have been deliberately removed. Such systematic absence of comments may reflect either manual editing practices or automated sanitization pipelines, potentially indicating institutional or departmental policies governing document preparation and public release.

To better understand the origins of these sanitized papers (approximately 4,600 in total), we applied *Named Entity Recognition (NER)* [16] to extract affiliation metadata, focusing on countries and organizations. The most frequent submitting countries were the United States (25%), China (15%), and Germany (7%). These proportions closely align with arXiv's own reported submission distribution [17], suggesting that no particular country appears to systematically enforce strict sanitization practices. At the organizational level, however, several large industrial research institutions were disproportionately represented among the submissions that showed evidence of comment removal, including Intel, Google, AWS, NVIDIA, IBM, Microsoft, and Apple.

Across the 92,000 remaining papers, we collected over 343,000 individual LaTeX source files containing over 12 million comments. These comments were passed through a multi-stage cleaning pipeline, which removed empty entries, boilerplate material, purely syntactic LaTeX commands, decorative separators, and redundancies. We also applied normalization steps to standardize whitespace and discarded

comments appearing more than 10 times across the dataset, as these primarily consisted of frequently used control sequences, formatting artifacts, or single words unlikely to contain sensitive information. After this pipeline, the dataset was reduced to approximately 6.6 million comments of varying length, representing just under 55% of the original set.

Dataset size reduction was especially important for enabling scalable LLM-based analysis. The raw LaTeX data set contained approximately 2.4 billion tokens under the `CL100k_base` encoding, while the cleaned comments amounted to fewer than 275 million tokens—an 89% reduction. For context, processing the unfiltered dataset with an LLM such as GPT-5-mini would cost more than $3000 at the input token rate of $1.25 per million tokens, not considering output tokens. By contrast, the cleaned dataset lowered inference costs to approximately $340 and significantly reduced inference time.

C. Stage 3 - Data Mining

In this step we use pattern-matching techniques, LLMs, and logical narrowing techniques to find sensitive information. As illustrated in Figure 1, this stage involves three main processes: pattern matching, LLM-based extraction, and logical filtering.

1) Pattern Matching: At this stage, pattern-matching techniques and regular expressions were applied to detect IP addresses, URLs, email addresses, banking information (such as IBANs), and other easily identifiable secrets in comments. The findings were then filtered and categorized manually and with LLMs. The remaining potentially sensitive links were separated into categories such as signed URLs, IP addresses, access tokens, and more. We utilized Trufflehog[2], a secret detection tool originally designed for Git repositories, to scan for API keys, user credentials, tokens, along with $1,750$ secret patterns from the *Secret Patterns Database Project* [18].

We excluded all information from the final results that clearly indicated it was a temporary or illustrative finding. Examples include loopback IP addresses, URLs, and email addresses using the *example.com* domain, and common tokens used for demonstration purposes. Running over 150 million regex searches on all extracted comments took approximately 66 minutes on a MacBook Pro M4.

2) Entity Extraction: LLMs were used to detect categories of sensitive information that could not be reliably captured through pattern-based techniques. Such information includes conversational content, for example, internal disagreements between co-authors, critical remarks concerning the validity or rigor of methods and results, as well as partial or full references to peer-review reports and corresponding rebuttals. Detecting these forms of sensitive text is particularly challenging without LLMs, as it requires contextual understanding and, often, multilingual comprehension.

To evaluate the effectiveness of LLMs for this task, we constructed a labeled dataset of 300 comment snippets: 200 containing sensitive information and 100 without. To emulate realistic conditions, we extracted content from 1,000 PDF files and embedded the sensitive snippets among them, reflecting the degraded performance observed when models must process longer inputs [13]. We refer to this dataset as `LLMsec-DB`, and it is available for download from the project website for further usage. Sensitive content was categorized into the following six classes:

- **PII**: Personally identifiable information such as names, email addresses, phone numbers, or physical addresses. This also includes hidden author names or other data that could reveal an individual's identity.
- **CRED**: Real authentication material or payment data. Examples include passwords or passphrases, API keys and tokens, bearer tokens, client secrets, private keys, database connection strings with embedded credentials, and payment card information (PAN, expiry, CVV/CVC/-CID).
- **NETID**: System or network identifiers used to label accounts or machines, but not secrets themselves. Examples include usernames, user IDs, IP addresses, hostnames, workstation IDs, MAC addresses, and port numbers.
- **PEER**: Content directly related to the formal peer review workflow, such as reviewer comments, meta-reviews, rebuttals, responses to reviewers, or camera-ready change requests. This also covers internal author discussions and strategy notes about how to respond to reviews, even if critical of reviewers.
- **CONF**: Explicit disagreements or disputes among co-authors, for example, regarding methodology, content, tone, style, or the direction of the paper. This category does not include disagreements with reviewers.
- **OTHER**: Miscellaneous sensitive content not covered by the above categories, or cases where no issue is present.

We benchmarked 25 different LLMs on this dataset to determine a balance between accuracy and computational cost. Since some snippets contained multiple categories of sensitive information, we report both *Exact-Match Accuracy (EMA)*, which requires all categories to be correctly identified, and *At-Least-One Accuracy (ONE)*, which indicates correct identification of at least one relevant label. For our purposes, ONE metric is the most relevant, as any positive match would trigger a manual review. Consider the following log snippet: *"Windows EventID 4672, ACCOUNT: sec_admin, PASSWORD: Pass!4682, MACHINE_ID: WEB_DEV-001*. If only one class of information is correctly identified—for example, the network identifiers but not the credentials—the case would still warrant further investigation and thus constitutes a valid finding in our analysis.

GPT-5 achieved the highest exact-match accuracy at 82.7%, while the top-performing open-source models were Qwen-2.5 72B and Llama-3.1 405B with 80% EMA. For partial accuracy (ONE), Gemini 2.5 Pro achieved the highest score at 93.3%, while Qwen-2.5 72B also reached a strong 91%, underscoring the competitiveness of open-source alternatives.

Cost considerations also play a significant role when scaling

[2]https://github.com/trufflesecurity/trufflehog

TABLE III: Performance of Tested LLMs on the LLM-SecDB Dataset (Sorted by Exact-Match Accuracy, EMA)

				CRED			PII			NETID			Costs	
Model	**EMA**	**ONE**	**O**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**TP**	**FP**	**FN**	**MT**	**ALL**
GPT-5	82.7%	91.7%	✗	50	3	0	42	16	2	38	9	4	$1.25	$840
Claude Sonnet 4	81.0%	87.7%	✗	50	0	0	44	10	0	39	8	3	$3.00	$2016
Claude 3.7 Sonnet	80.3%	92.0%	✗	50	0	0	43	21	1	37	12	4	$3.00	$2016
Qwen-2.5 72B	80.0%	91.0%	✔	50	2	0	44	11	0	38	11	3	$0.07	$47
Llama-3.1 405B	80.0%	87.7%	✔	49	2	1	44	11	0	36	10	5	$0.80	$538
Lllama-3.3 70B	76.0%	86.3%	✔	49	5	1	44	7	0	35	12	6	$0.04	$27
Gemini 2.5 Flash Lite	77.0%	86.0%	✗	45	1	5	44	8	0	33	10	8	$0.10	$67
GPT-5 Nano	76.7%	85.0%	✗	50	6	0	44	17	0	34	13	7	$0.05	$34
Gemini 2.5 Pro	75.0%	93.3%	✗	50	0	0	42	24	2	32	14	8	$1.25	$840
Deepseek R1	75.0%	86.7%	✔	48	5	2	43	17	1	31	16	9	$0.40	$269
GPT-5 mini	71.3%	84.3%	✗	50	11	0	44	24	0	30	17	10	$0.25	$168
NVIDIA Nemotron 9B	70.3%	80.3%	✔	50	15	0	44	19	0	29	18	11	$0.25	$168
GPT-4o mini	67.7%	86.3%	✗	50	13	0	43	9	1	28	17	13	$0.15	$101
Grok4	67.0%	81.7%	✗	49	0	1	42	20	2	27	20	14	$3.00	$2016
Qwen3 235B-A22b	78.7%	86.0%	✔	48	3	2	42	2	2	33	8	8	$0.30	$202
Gemma 3 27B	65.7%	82.0%	✔	50	18	0	44	25	0	27	19	15	$0.07	$47
Gemini Flash-2.5	65.0%	74.3%	✗	48	10	2	43	5	1	25	20	17	$0.30	$202
Mixtral 8x22b	63.3%	77.3%	✔	43	3	7	44	25	0	22	24	20	$0.90	$605
Mixtral 8x7b	54.0%	65.3%	✔	38	2	12	31	6	13	21	25	21	$0.40	$269
Qwen2.5 7B	54.7%	65.7%	✔	47	26	3	36	24	8	20	26	22	$0.04	$27
Gemma-3 12B	55.7%	80.0%	✔	50	27	0	44	24	0	19	27	23	$0.04	$27
Claude-3.5 Haiku	56.0%	79.0%	✗	50	12	0	44	42	0	18	28	24	$0.08	$54
Mistral 7B	59.0%	76.7%	✔	46	11	4	40	13	4	17	29	25	$0.03	$20
Llama-4 Scout	59.0%	80.0%	✔	50	43	0	44	20	0	16	30	26	$0.08	$54
Ministral 3B	49.3%	56.7%	✔	37	2	13	23	11	21	14	32	28	$0.04	$27

Legend: **EMA** = Exact-match accuracy; **ONE** = At least one correct. **O** = Open-source;
MT = Cost per million input tokens; **ALL** = Approximate cost for all 100,000 articles;

to our full dataset. According to OpenRouter.ai pricing, GPT-5 input tokens are priced at $1.25 per million, whereas Qwen-2.5 72B costs only $0.07—approximately 96% lower, with even higher savings for output tokens. Given that the cleaned volume of comments across 100,000 papers contains approximately 224 million input tokens, the estimated cost for Qwen-2.5 72B is about $15.68, compared to over $280 for GPT-5, excluding output tokens or cached system prompts, which roughly triples the costs. The evaluated LLMs and their results on the `LLMSec-DB` benchmark with 300 entries are presented in Table III. We note that TruffleHog identified only four instances of real credentials in our dataset while producing an exceptionally high number of false positives. As a result, its EMA and ONE scores were below 10%, clearly demonstrating that it is unsuitable for secret detection beyond simple pattern matching. Based on these findings, we selected Qwen-2.5 72B for the large-scale arXiv submission analysis, as it achieved exceptionally high EMA and ONE scores on our benchmark while also offering cost-efficient inference. Using parallel execution through the OpenRouter.ai API, Qwen-2.5 72B processes the small `LLMsec-DB` dataset of 300 entries in 15–20 seconds. However, analyzing the full set of LaTeX files with comments requires more than 5 hours and incurs a cost of approximately $50.4 (including the preprompt and output tokens). All outputs were stored in a structured JSON format for further analysis and manual validation if necessary. Illustrative cases are presented in Section IV to highlight the typical findings of this study while ensuring that the identities of individual authors remain undisclosed.

3) Logical Filtering: Non-LaTeX source files that might contain sensitive information were identified through logical filtering. We began by mapping every file referenced in the compilation of the final PDF. This process yielded a set of files explicitly intended to appear in the published document. These files were excluded, and we retained only those that were never imported into the final output. Such unreferenced files may exist for several reasons:

1) The file served as a temporary placeholder or was superseded by a newer version,
2) The file was used as an internal guide for formatting or content reference,
3) The file originated from collaborative platforms (e.g., Overleaf) and includes measurements, discussions, or procedural notes not intended for publication

Our primary focus was on files of the third category, as these are most likely to contain unpublished or sensitive material. Subsequently, we filtered out files deemed unlikely to expose sensitive data, including chart and figure sources (e.g., `.eps`), bibliographic files (`.bib`, `.bbl`), styling and rendering artifacts (`.pygtex`), and similar. After this step, the dataset was reduced to approximately 70,000 candidate files. These consisted of data files (`.xml`, `.json`, `.csv`, `.html`), plain text documents (`.txt`, `.md`, `.list`), source code (`.py`, `.js`), logs and backups (`.log`, `.bkp`, `.db`), among others. In this stage, we reapplied techniques similar to those in Stage 1. Specifically, we used TruffleHog for secret detection together with 1,750 regular expressions from the Secret Patterns Database, and we performed metadata

(EXIF) extraction to recover location, timestamp, authoring information, and editing history. All detected matches were subsequently subjected to manual validation to eliminate false positives.

D. Stage 4 - Analysis

In the final stage, we implemented automated filtering pipelines to reduce false positives, followed by manual validation of the remaining candidates. The extracted information, stored in `.jsonl` files, was systematically categorized and analyzed using Python scripts to generate statistical summaries in the form of charts and tables. Particular attention was devoted to categories of data deemed especially sensitive, such as login credentials and records exposing significant amounts of PII, which were manually reviewed in detail.

During the manual verification process, we identified several critical cases of exposed information, including instances of private authentication data. When such exposures were confirmed, we attempted to locate contact information and directly notified the affected authors. To protect the privacy of researchers and prevent data misuse, all analyzed outputs were anonymized and sanitized. The dataset was further processed to ensure that the results could not be reverse-engineered to reconstruct the exact set of archives included in our scans. A more detailed account of these findings and their implications is presented in Section IV.

IV. Discussion & Results

In this section, we present the key findings from our large-scale analysis of 100,000 arXiv submissions. The analysis can be divided into two distinct research directions: the first relies on traditional tools and techniques such as TruffleHog to detect secrets in comments and files, while the second leverages advanced LLMs. A key observation is that, apart from simple pattern matching—which proved useful for identifying IP addresses and URLs—TruffleHog was largely unsuccessful in detecting genuine secrets in comments and files. Most of its outputs were false positives. For example, credentials mentioned in a paper's discussion of weak passwords were incorrectly flagged as leaks. This is expected, as such tools cannot interpret the surrounding text or contextual meaning. Our manual verification confirmed that traditional secret-leak detection methods yield very few, if any, meaningful findings. By contrast, the LLM-based analysis produced exceptionally strong results, revealing real credential leaks, internal sensitive author discussions, PII, and other sensitive data. In this section, we systematically present the most significant and illustrative findings.

A. Conventional Methods of Secret Leak Detection

Using simple pattern-matching techniques and regex searches, we extracted approximately 42,500 unique URLs, including IP addresses, from the LaTeX comments. We performed ISP and domain reverse lookups on all collected addresses and found that 22% originated from the United States, 9% from France, and 7% from China, with the top ISPs being Orange, Level3, and Amazon. The most common domains are presented in Figure 3.

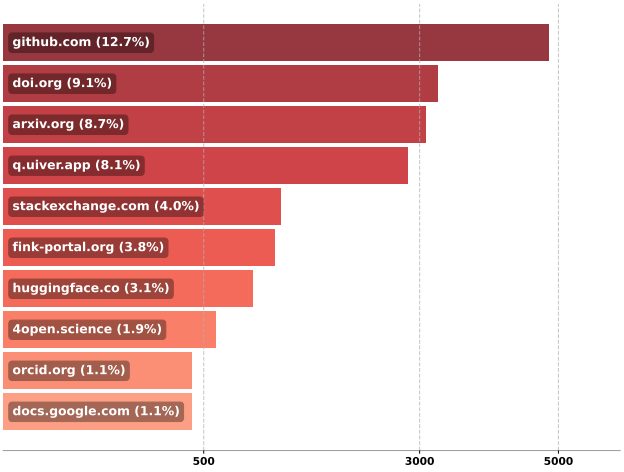

Fig. 3: The ten most commonly occurring domains in URLs extracted from the comments.

Although these domains are noteworthy, they rarely contain sensitive information. Instead, such data is more commonly found in file-sharing portals or private websites, which in some cases expose access tokens directly within the URL. We identified 206 IP addresses—only eight in private ranges—with ports exposing web servers, databases, FTP (20/21), SSH (22), unencrypted web services (80, 8080, 8000, 4321), printers (9100), and other open services. Even more concerning, we discovered a private online chat platform in which login credentials were embedded in the URL, effectively making the entire chat history publicly accessible.

We also identified over 700 links to websites that have a long token-like path or query parameters. This indicates that many of these might grant access to the resources without further authentication. Using simple pattern matching, we collected more than 30 CASA tokens, which grant free access to scientific publications. These tokens are typically issued when students or researchers with institutional subscriptions share links internally, and they are not intended to be publicly exposed. Furthermore, we identified nearly 650 links granting view or edit access to private files and folders. These links belong to popular cloud storage services, including Google Drive, Dropbox, Proton Drive, Mega, and others. About 70% of the content could be accessed without additional authentication. Leaked documents can be categorized into the following groups:

- Peer-review materials, including reviews, rebuttals, and associated discussions,
- Internal correspondence reflecting reviewer or author skepticism about results,
- Links to internal repositories and development environments accessible via direct URL,
- Draft manuscripts of other works, some containing embedded comments,

- Unreleased training datasets (text, image, multimodal) and unpublished trained model weights,
- Documents containing additional signed links that enable lateral access to related resources,
- Spreadsheets with experimental measurements, including instances that differ from those reported in the published article,
- Survey data, in some cases mapping responses to individual email addresses,
- Spreadsheets or tables containing PII,
- Private letters, future research plans, internal documentation, and instructional guides, as well as supplementary media such as images, videos, and charts.

It is also important to highlight that around 85% of the shared documents also allow editing, as they were likely shared by researchers for easy collaboration. At this stage, we identified multiple categories of sensitive information, including several AWS secret access keys, more than 90 IBANs, P.O. box addresses, institutional addresses, and phone numbers that, based on manual verification, were not intended for publication.

B. LLM-Based Methods of Secret Leak Detection

The LLM-based analysis produced far more sophisticated and serious results. As described in the methodology section, we employed Qwen-2.5 72B to identify hidden secrets, leaked credentials, and internal author discussions that could not be detected through simple pattern matching. These represent some of the most interesting findings, as such discoveries were not feasible before the advent of LLMs. They require contextual understanding of text, a task at which current LLMs excel. The QWEN-2.5 72B model flagged 9,926 papers, with some receiving multiple flags, resulting in a total of 13,806 detections. The five primary categories (excluding the OTHER category) are shown in Table IV.

PII, conflict, and peer-review–related findings dominate the distribution, though it is notable that 47 credentials were also identified in the articles. We note that the CRED category refers strictly to authentication secrets (e.g., passwords). Tokens embedded in URLs, including those granting access to shared private folders, are handled separately under sensitive URLs (c.f. pattern matching of URLs). We manually validated all credentials and network identifiers and further validated around 10% each of the conflicts, peer reviews and PII. Table V presents anonymized real-world credential leakages from comments, as identified by QWEN-2.5 72B. All such findings are manually reviewed and verified.

In numerous instances, the exposed credentials were accompanied by the original login page URL, further escalating the risk by revealing exactly where an attacker could attempt to use them. This analysis highlights that, for as little as $50 using LLM-based analysis—in our case Qwen-2.5 72B—a malicious actor could obtain valuable information.

V. Limitations and Threats to Validity

We have identified the following limitations of our study:

TABLE IV: Distribution of Label Categories

Category	Count
PII	5689
CONF, PEER	2228
PEER, PII	518
CONF	383
CONF, PII	331
CONF, PEER, PII	330
PEER	294
NETID, PII	71
CRED, PII	37
NETID	17
CRED	10
CRED, NETID, PEER	5
NETID, PEER	4
CRED, NETID, PEER, PII	4
NETID, PEER, PII	3
CRED, NETID	1
CRED, NETID, PII	1
Overall Distribution by Category (summarized)	
PII	6984
PEER	3386
CONF	3283
NETID	106
CRED*	47
SUM	**13806**

TABLE V: Sanitized examples of credential leaks in arXiv submissions.

ID	Type	Sanitized Example
1	Portal Login	`https://****.org`, Paper ID: `****`, Password: `[*****]`
2	Email + password	Email: `w*****@p****.org.cn`, Password: `*****`
3	Project website	`https://sso.****.com`, Email: `****@****.com`, PW: `[*****]`
4	Priv. SharePoint link	`https://unsw-my.sharepoint.com/:u:/g/personal/[****]`
5	Web Login	Login: `https://****/login`, Username: `reviewer`, Password: `[*****]`
6	Github account	**** `github.com/******` pass: [[*****]]
7	Gmail account	**** `l*****gmail.com` pass: [[*****]]
8	Paper submission	Your paper number is: ***4. Your paper access password is: C********
9	Private docs	`https://docs.google.com/doc//d/1jC2****ULw/edit`

Orange = moderately sensitive data; Red = critically sensitive data.

- The `LaTeXpOsEd` framework is designed around LaTeX-based source submissions and their associated auxiliary files. Disclosures that occur exclusively in non-LaTeX workflows—for example, manuscripts prepared in Word or platforms that encapsulate all assets directly into PDFs—are not captured in our analysis.
- The accuracy of our detection pipeline is constrained by the limitations of both pattern-matching heuristics and LLMs used for entity extraction. This includes risks of false positives on contextually benign strings, as well as false negatives when secrets are obfuscated or encoded in non-standard forms. Encoded formats such as Base64

TABLE VI: Summary of Sensitive Information Disclosures by Severity and Frequency

Description	Source	Detection	Count*	Severity	Impact
Login credentials (username/password)	comments	EE	47	Critical	System compromise
Large-scale (aggregated) PII data exposure	links	PM	22	Critical	Mass privacy breach
Survey data with large-scale PII exposure	links	PM	2	Critical	Mass privacy breach
AWS access keys	comments	PM	2	Critical	Full infrastructure compromise
PII Exposure	comments	EE	6984	High	Privacy breach
Author conflicts and admission of false results	comments	EE	3283	High	Reputational damage
Image location EXIF data	files	LF	631	High	Privacy breach
Private documents and folders	links	PM	129	High	Data leakage
JSON Web Tokens (JWTs) without expiry date	comments	PM	5	High	Account hijack
Peer reviews disputes / disagreements	comments	EE	3386	Medium	Reputational damage
Private source code	links, files	PM, LF	2774	Medium	Data leakage
Private Git repository addresses (HTTP/SSH)	comments	PM	658	Medium	Code and data exposure
Unintended Public Exposure of IP addresses	comments	PM	198	Medium	Network mapping and targeting
Phone numbers (not publicly accessible by design)	comments	PM	60	Medium	Spam/targeted phishing risk
CASA tokens (publication access)	comments	PM	30	Medium	Financial damage
Private model weights and training data	links, files	PM, LF	28	Medium	Data leakage
Social security numbers (US, UK)	comments	PM	26	Medium	Identity theft risk
SQLite databases	files	LF	7	Medium	Data leakage
Unique email addresses	comments	PM	8236	Low	Spam/targeted phishing risk
Structured configuration files and metadata (JSON, YAML)	files	LF	802	Low	Data exposure
Hashes with unknown content (MD5, SHA1, SHA256)	files, comments	LF, PM	549	Low	Misc data
Network & Device Identifiers (e.g., MAC, Host ID),	comments	EE	106	Low	Network mapping and targeting
Validated IBANs	comments	PM	92	Low	Financial data exposure
P.O Box addresses	comments	PM	75	Low	Privacy breach

Legend: **PM** = Pattern Matching; **EE** = Entity Extraction; **LF** = Logical Filtering
* Counts obtained through entity extraction are approximate; based on our benchmark evaluation, their accuracy is within a margin of $\pm 9\%$.

or hexadecimal are not handled efficiently.

- At the time of this study, arXiv hosted nearly three million submissions, of which our dataset represents approximately 3.6%. While this proportion is broadly representative, generalizations to the full corpus should be interpreted as illustrative rather than definitive.
- The financial cost of large-scale analysis also represents a significant limitation. While processing 100,000 submissions required approximately $90, extrapolating to the full arXiv archive yields an estimated cost of around $25,000. This calculation excludes additional expenses such as server infrastructure and human validation time, which vary considerably. As such, the approach remains prohibitively expensive for most potential actors.

VI. Conclusion

In this study, we introduced the `LaTeXpOsEd` framework, a novel methodology that integrates modern artificial intelligence techniques with traditional analysis to identify and characterize sensitive information leaks in public preprint archives. Our primary objective was to illustrate both the scale of inadvertent disclosures and the risks arising from unsanitized article submissions. By applying this framework to a representative sample of 100,000 arXiv submissions, we demonstrated that sensitive data leakage is a significant and underappreciated concern.

Our analysis uncovered hundreds of instances of exposed credentials granting access to private accounts and documents, thousands of cases of unintentional disclosure of peer review materials and author correspondence, and numerous examples of internal author communications, including critical discussions about methodology and validity. We also identified disclosures of PII, contact details, and metadata embedded in auxiliary files. These findings underscore that the risks are not hypothetical but widespread, extending across disciplines and institutions. The findings are summarized in Table VI.

Based on our findings, we emphasize the need for strong prevention measures at both the author and platform levels. For authors, this includes incorporating systematic sanitization practices before submission. Authors should:

- Use freely available tools such as *arXiv LaTeX Cleaner*[3] to remove comments and extraneous files before submission;
- Avoid uploading credentials, sensitive documents, or notes to collaborative editing platforms (e.g., Overleaf), and instead rely on secure version control or shared storage;
- Refrain from disseminating cloud storage materials via signed URLs with edit permissions, and instead share access directly and privately with co-authors.
- Use separate credentials for research projects (e.g., dedicated API keys with minimal scope) to limit potential damage if leaks occur.
- Regularly audit project directories to ensure auxiliary files (e.g., .log, .aux, .bbl, temporary caches) do not contain sensitive data.

For preprint platforms, stronger systemic safeguards are equally critical. We recommend:

- Providing clear warnings to authors that all files included in source submissions are made publicly available and cannot be retracted once published,

[3] https://github.com/google-research/arxiv-latex-cleaner

- Implementing automated sanitization procedures (e.g., removal of comments and unused files), and
- Considering the option of withholding original source packages from public release altogether.
- Providing integrated credential and sensitive-data scanners (e.g., detecting API keys, tokens, or email addresses) at upload time, with immediate feedback to the author;

Adopting these measures would significantly reduce the risk of unintentional disclosures in the future. However, our analysis highlights risks under systemic, large-scale conditions, while a determined adversary targeting an individual researcher's complete submission history could, with sufficient effort, uncover additional proprietary or confidential information beyond the reach of our automated pipeline.

Finally, our study addressed two guiding research questions:

RQ1: What kinds of sensitive information are exposed in preprint source files and comments, and how often do these disclosures occur?
Answer: Approximately 10% of papers in our 100,000-paper sample contained sensitive disclosures, including PII, internal communications, confidential peer-review documents, IP addresses, URLs, and image metadata. Around 0.2% of papers included information classified as *critical*, such as login credentials, API tokens, and indications of fake research data.

RQ2: How effective are traditional pattern-matching techniques and LLMs at detecting contextually hidden sensitive data within preprint source materials?
Answer: Both approaches demonstrated strengths, with pattern matching and logical filtering being effective at detecting structured artifacts such as URLs, while LLMs significantly reduced false positives by leveraging contextual understanding. For example, LLMs successfully identified credentials based on surrounding context, detecting 42 true positives in our validation procedure, compared to 4,700 false positives flagged by TruffleHog. LLMs are, however, considerably more expensive than pattern matching and logical filtering.

Our work highlights the urgent need for stronger community awareness, improved author practices, and systematic interventions at the platform level to ensure that the benefits of open scientific communication are not undermined by preventable risks to privacy and security.

Acknowledgement

This research is supported and funded by ZEISS Digital Innovation and the Technology Innovation Institute (TII), Abu Dhabi. Additional support is provided by the TKP2021-NVA Funding Scheme under Project TKP2021-NVA-29; ELTE-OTP Cyberlab—a collaboration between Eötvös Loránd University (ELTE) and OTP Bank Plc; EPSRC grant EP/T026995/1 titled "EnnCore: End-to-End Conceptual Guarding of Neural Architectures" under the Security for All in an AI-enabled Society program; the Research Council of Norway Project No. 312122 "Raksha: 5G Security for Critical Communications"; funding from Horizon Europe under Grant Agreement No. 101120853; and funding under Grant Agreement No. 101145874, supported by the European Cybersecurity Competence Centre.

References

[1] National Science Board, "Publication output by country, region, or economy and scientific field," 2021, accessed: 2025-09-25. [Online]. Available: https://ncses.nsf.gov/pubs/nsb20214/publication-output-by-country-region-or-economy-and-scientific-field
[2] J. Pastor-Galindo, P. Nespoli, F. Gómez Mármol, and G. Martínez Pérez, "The not yet exploited goldmine of osint: Opportunities, open challenges and future trends," *IEEE Access*, vol. 8, pp. 10 282–10 304, 2020.
[3] B. Butler, B. Wardman, and N. Pratt, "Reaper: an automated, scalable solution for mass credential harvesting and osint," in *2016 APWG Symposium on Electronic Crime Research (eCrime)*, 2016, pp. 1–10.
[4] J. Pei, L. Yang, and L. Wu, "Hidden division of labor in scientific teams revealed through 1.6 million latex files," *arXiv preprint arXiv:2502.07263*, 2025.
[5] T. Saier and M. Färber, "unarxive: a large scholarly data set with publications' full-text, annotated in-text citations, and links to metadata," *Scientometrics*, vol. 125, no. 3, pp. 3085–3108, 2020.
[6] B. Hepler, "Modular versus hierarchical: A structural signature of topic popularity in mathematical research," 2025. [Online]. Available: https://arxiv.org/abs/2506.22946
[7] P. Scharpf, M. Schubotz, A. Youssef, F. Hamborg, N. Meuschke, and B. Gipp, "Classification and clustering of arxiv documents, sections, and abstracts, comparing encodings of natural and mathematical language," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, 2020, pp. 137–146.
[8] M. Cheng, K. Gligoric, T. Piccardi, and D. Jurafsky, "Anthroscore: A computational linguistic measure of anthropomorphism," *arXiv preprint arXiv:2402.02056*, 2024.
[9] M. L. Bianchi, "Text mining arxiv: a look through quantitative finance papers," 2024. [Online]. Available: https://arxiv.org/abs/2401.01751
[10] A. T. A, C. Kumar, A. Amal, and A. Kumar, "Ai academic research aggregator," in *2025 3rd International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*, 2025, pp. 1–5.
[11] N. Meuschke, A. Jagdale, T. Spinde, J. Mitrović, and B. Gipp, "A benchmark of pdf information extraction tools using a multi-task and multi-domain evaluation framework for academic documents," in *Information for a Better World: Normality, Virtuality, Physicality, Inclusivity*, I. Sserwanga, A. Goulding, H. Moulaison-Sandy, J. T. Du, A. L. Soares, V. Hessami, and R. D. Frank, Eds. Cham: Springer Nature Switzerland, 2023, pp. 383–405.
[12] N. Tihanyi, T. Bisztray, R. A. Dubniczky, R. Toth, B. Borsos, B. Cherif, R. Jain, L. Muzsai, M. A. Ferrag, R. Marinelli *et al.*, "Dynamic intelligence assessment: Benchmarking llms on the road to agi with a focus on model confidence," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024, pp. 3313–3321.
[13] R. A. Dubniczky, K. Z. Horvát, T. Bisztray, M. A. Ferrag, L. C. Cordeiro, and N. Tihanyi, "Castle: Benchmarking dataset for static code analyzers and llms towards cwe detection," in *International Symposium on Theoretical Aspects of Software Engineering*. Springer, 2025, pp. 253–272.
[14] M. N. Rahman, S. Ahmed, Z. Wahab, S. M. Sohan, and R. Shahriyar, "Secret breach detection in source code with large language models," 2025. [Online]. Available: https://arxiv.org/abs/2504.18784
[15] J. Tesic, "Metadata practices for consumer photos," *IEEE MultiMedia*, vol. 12, no. 3, pp. 86–92, 2005.
[16] I. Keraghel, S. Morbieu, and M. Nadif, "Recent advances in named entity recognition: A comprehensive survey and comparative study," 2024. [Online]. Available: https://arxiv.org/abs/2401.10825
[17] J. Entwood, "arxiv hits 12k in may," arXiv Blog, May 2018, accessed 2025-09-17. [Online]. Available: https://blog.arxiv.org/2018/05/31/arxiv-hits-12k-in-may/
[18] M. Ahmed, "Secrets patterns db: The largest open-source database for detecting secrets, api keys, passwords, tokens, and more," GitHub repository, 2023, cC-BY-SA-4.0 license. [Online]. Available: https://github.com/mazen160/secrets-patterns-db