Journal of Software Engineering and Applications, 2024, 17, 421-447
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Evaluating Privacy Leakage and Memorization Attacks on Large Language Models (LLMs) in Generative AI Applications

Harshvardhan Aditya[1], Siddansh Chawla[1], Gunika Dhingra[1], Parijat Rai[1], Saumil Sood[1], Tanmay Singh[1], Zeba Mohsin Wase[1], Arshdeep Bahga[2], Vijay K. Madisetti[3]

[1]School of Computer Science Engineering & Technology, Bennett University, Greater Noida, India
[2]Cloudemy Technology Labs, Chandigarh, India
[3]School of Cybersecurity and Privacy, Georgia Institute of Technology, Atlanta, USA
Email: harshva27@gmail.com, chawlasiddansh@gmail.com, 12gunika@gmail.com, parijatrai02@gmail.com, sood.saumil03@gmail.com, singhtanmay0915@gmail.com, zeba.wase@gmail.com, arshdeep@cloudemy.io, madisetti.vijay@gmail.com

How to cite this paper: Aditya, H., Chawla, S., Dhingra, G., Rai, P., Sood, S., Singh, T., Wase, Z.M., Bahga, A. and Madisetti, V.K. (2024) Evaluating Privacy Leakage and Memorization Attacks on Large Language Models (LLMs) in Generative AI Applications. *Journal of Software Engineering and Applications*, **17**, 421-447.
https://doi.org/10.4236/jsea.2024.175023

Received: April 20, 2024
Accepted: May 28, 2024
Published: May 31, 2024

Abstract

The recent interest in the deployment of Generative AI applications that use large language models (LLMs) has brought to the forefront significant privacy concerns, notably the leakage of Personally Identifiable Information (PII) and other confidential or protected information that may have been memorized during training, specifically during a fine-tuning or customization process. We describe different black-box attacks from potential adversaries and study their impact on the amount and type of information that may be recovered from commonly used and deployed LLMs. Our research investigates the relationship between PII leakage, memorization, and factors such as model size, architecture, and the nature of attacks employed. The study utilizes two broad categories of attacks: PII leakage-focused attacks (auto-completion and extraction attacks) and memorization-focused attacks (various membership inference attacks). The findings from these investigations are quantified using an array of evaluative metrics, providing a detailed understanding of LLM vulnerabilities and the effectiveness of different attacks.

Keywords

Large Language Models, PII Leakage, Privacy, Memorization, Overfitting, Membership Inference Attack (MIA)

1. Introduction

Large Language Models (LLMs) are widely used in Generative AI applications

that are increasingly being deployed in a variety of business processes at scale. These LLMs, trained on vast amounts of data, excel at understanding and processing text similarly to humans. They are capable of performing various natural language tasks such as text generation, translation, and sentiment analysis [1]. Owing to their complexity, LLMs contain millions to billions of parameters and can be applied in diverse fields including technology, healthcare, marketing, legal, and banking.

Pre-trained models (or Foundation LLMs) are versatile and can be used across various business domains, but integrating them into specific systems for enhanced or targeted performance requires re-training or specialized training, known as fine-tuning [2]. This process adjusts pre-trained models to specific tasks or datasets, improving their accuracy and performance in particular areas. Fine-tuning updates the model's parameters using new data or examples, allowing it to specialize to more focused tasks while maintaining its general language abilities. Different fine-tuning approaches include supervised learning with labeled datasets, domain-specific adjustments for better performance in certain industries, and transfer learning, which applies general knowledge to more focused tasks.

After fine-tuning, an LLM's ability to answer questions based on a specific dataset greatly improves, but it often retains much of the information it received during the process. This capability, known as memorization [3] [4], allows the model to remember and sometimes reproduce exact information from its training data when a user queries (or "prompts") it appropriately. Research [5] indicates that as LLMs grow larger with more parameters, memorization becomes more common unless steps are taken to prevent it. Memorization can also progress to over-fitting [6], where a model while performing very well on its training data may also store information that when prompted to recall specific details, reveals sensitive, confidential, or private information that was present in the training data set.

A major issue arises when training datasets for Large Language Models (LLMs) contain sensitive and confidential information, such as Personal Identifiable Information (PII) [7]. The leakage of PII is a significant concern because it can directly or indirectly identify individuals, which is regulated under laws like the General Data Protection Regulation (GDPR) [8]. PII includes direct identifiers such as names, addresses, and Social Security numbers, and indirect identifiers like gender, race, and birthdate. Thus, the challenge in generative AI applications is to balance the advantages of LLMs with the need to protect PII and other confidential information. When fine-tuning LLMs and creating specific language models for specialized business domains, it is important to consider all privacy and confidentiality issues related to the chosen techniques, models, datasets, and other parameters. This consideration is essential to protect user data and ensure the integrity of the model's output.

In light of all of these factors, under certain conditions, LLMs may pose significant security and privacy risks, potentially impacting individuals and organ-

izations by enabling data breaches or ransomware attacks when attacked by malicious parties. These models can also bypass security systems or generate harmful content, undermining digital safety. Such risks, often linked to the models' memorization abilities, raise significant concerns about privacy, effectiveness, and fairness.

We describe and test various adversarial attack scenarios on LLMs to evaluate their effectiveness using detailed performance metrics. We also conduct experiments using LLMs of varying sizes and types to investigate how the number of parameters influences memorization, leakage, and eventually even the potential leakage of personally identifiable information (PII) and other confidential information. We have also developed a user-friendly plug-and-play web interface that allows easy access and interaction with the models. This platform enables users to execute different attacks and helps in understanding the extent of memorization, providing a broad overview of the intersection between language models and security.

The key contributions of this paper are:

- Analysis of two important types of black-box LLM attacks: PII leakage-focused attacks (*i.e.*, auto-completion and extraction attacks) and memorization-focused attacks (*i.e.*, membership inference attacks).
- Investigation of the influence of the types and sizes of LLM models on memorization, leakage, and potential PII exposure under black-box attacks through measurement of a set of metrics.
- Development of a user-friendly web interface that allows easy access and interaction with the models, enabling users to execute different attacks and understand the extent of memorization.

2. Related Work

Herein, we examine some recent results by other authors that provide some of the background and foundation for our work.

1) Practical Membership Inference Attacks against Fine-tuned Large Language Models via Self-prompt Calibration: This study [9] investigates how fine-tuned large language models (LLMs) are vulnerable to membership inference attacks (MIAs), including both reference-free and reference-based types. The research introduces a new black-box attack method, Self-calibrated Probabilistic Variation (SPV), which leverages the phase of memorization during fine-tuning to extract confidential information from the LLM. This attack also involves using a classifier to identify if a sample was in the training data of the model.

2) Empirical Analysis of Memorization in Fine-tuned Autoregressive Language Models: This study [5] suggests that fine-tuned LLMs significantly memorize training data, which decreases their performance on new, and unseen data. This memorization, according to the authors, is especially notable when fine-tuning with smaller datasets. Further, this study finds that fine-tuning of certain types (e.g., the head adapter layer of the LLM) may increase vulnerability

to privacy attacks due to increased memorization.

3) Analyzing Leakage of Personally Identifiable Information in Language Models: This study [10] examines the risk that LLMs, like GPT-2, leak Personally Identifiable Information (PII) when fine-tuned in domains such as case law, healthcare, and email. It defines PII leakage attacks that generating exact text sequences, reconstruct PII from masked training data sentences, and infer PII using a target sequence and candidate PII set, respectively. The findings also suggest that while aggressive data scrubbing and differential privacy may enhance security and privacy, they inversely affect the LLMs utility.

4) Quantifying Memorization across Neural Language models: This study [4] analyzes how as the LLMs capacity increases, the use of replicated training data examples and added contextual tokens increase memorization in LLMs across different models and datasets. It focuses on certain types of extraction attacks, where the use of replication in training data leads to more data leakage [11] due to increased memorization. The study shows that even querying a small portion of the training data provides reliable estimates of leakage. The results also indicate that larger models memorize more that smaller models.

5) Does Fine-tuning GPT-3 With the OpenAI API Leak Personally-Identifiable Information?: This study [12] also explores the privacy risks of fine-tuned large language models like GPT-3, focusing on the dangers of leaking Personally Identifiable Information (PII). The researchers used OpenAI's fine-tuning API to evaluate the LLM's vulnerability to PII extraction. Their findings reveal that fine-tuning for tasks such as classification and auto-complete can cause GPT-3 to memorize and disclose sensitive PII from its training data when prompted appropriately. Both naive (classification) and practical (auto-complete) settings are shown to be vulnerable allowing users with just a black-box API access to extract PII using appropriate prompts.

In this paper, we build on this exciting recent work, and extend their analyses.

3. Overview of Our Paper

This paper seeks to advance the understanding of security and privacy in Large Language Models (LLMs) by applying sophisticated attacks to carefully selected representative LLMs and specialized fine-tuning datasets. As described in **Figure 1**, we evaluated the performance of LLMs under two categories of privacy attacks, and utilized several types of datasets and LLMs to measure suitable metrics. We briefly describe the LLMs attacked, the datasets used in fine-tuning, the types of attacks, and metrics used, in the sections that follow.

3.1. LLMs Tested and Attacked

Several LLMs were utilized in our study as fine-tuned target and reference models. These models vary in size, architecture and security measures, offering a broad spectrum of capabilities and specializations. Each model is briefly described below, including their parameter count and distinctive features.

- **LLaMA 7B**: A foundational language model with **7 billion** parameters [13],

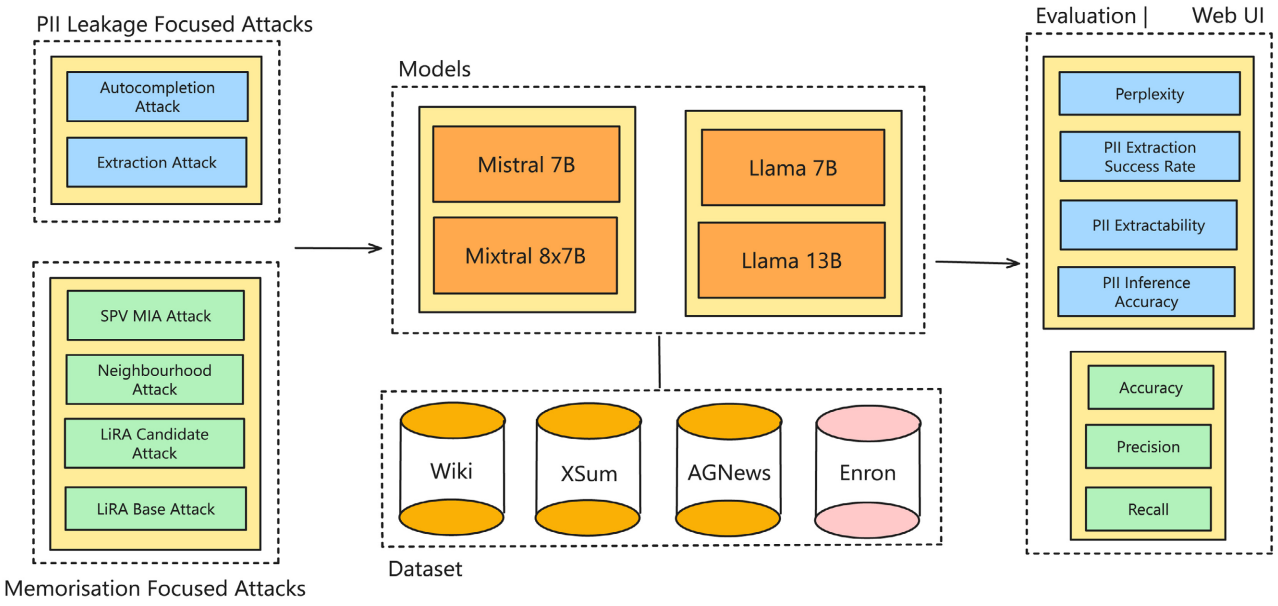

Figure 1. Overview of evaluation workflow used in paper.

trained on a trillion tokens. It excels in that tasks of dialogue generation and role-play despite its smaller size.

- **Mistral 7B**: A **7.3 billion** parameter language model [14], noted for superior performance in English language tasks.
- **LLaMA 13B**: Known for efficiently managing various domains with **13 billion** parameters [13], this model competes well with larger models.
- **Mixtral 8x7B**: A **45 billion** parameter model using a so-called "mixture of experts" architecture [15]. During inference, it selects two best models and uses approximately **12.9 billion** parameters to obtain its performance.

3.2. Datasets Used for Fine-Tuning LLMs

We now describe the datasets employed in our evaluation workflow. These datasets are classified according to their use in various types of attacks, such as inference and memorization attacks, and further divided based on their application in training either target or reference models.

3.2.1. Fine-Tuning Datasets

This section briefly details the datasets utilized in fine-tuning of the LLMs in this study.

- **Datasets for PII Leakage Focused attacks:**
- **Enron-Body Dataset:** Collection of the text content in the body of emails from an Enron dataset [16] [A collection of real-world business emails from the Enron Corporation].
- **Enron-Prompt Template Dataset:** Includes both the email subject and body text.
- **AESLC (Annotated Enron Subject Line Corpus) Dataset:** Used as the basis to prepare our prompt template dataset:

https://huggingface.co/datasets/aeslcAESLC. The dataset preparation was done in a similar way to [12].

- **Datasets Used for Memorization Focused Attacks:**
- **Wikitext:** A collection of Wikipedia articles formatted for natural language processing.
- **AG News:** A collection of news articles categorized into four main topics.
- **XSum:** A collection of summaries of BBC articles for text summarization tasks.
- **Data Processing for Target Models:**
- Memorisation Datasets & Enron Body Dataset: Train/test split of 10,000 training and 1000 testing samples, random sampling with seed 42.
- Enron Prompt Template Dataset: Train/test split of 500 training and 100 testing samples, employing instructive prompt templates and model types.

3.2.2. Reference Datasets

These datasets were selected as they are the non-overlapping but similar domain specific datasets for the memorization attack datasets respectively.

Dataset Info:

- **Wikicorpus:** A large corpus of multilingual Wikipedia articles for linguistic research.
- **TLDR News:** Summarized news headlines for quick consumption.
- **CNN Daily Mail:** Dataset containing news articles and their summaries from CNN and Daily Mail.

Data Processing for Reference Models:

- Train/test split of 1000 training and 200 testing samples, random sampling with seed 42.

Used for: Fine-tuning reference models for the LiRA Candidate Attack and SPV-MIA attack (using 8 words of prompt input from same reference dataset).

3.2.3. Evaluation Datasets

Data Pre-processing:

- **Member Data (500 Members):** Randomly selected from one of the memorisation attack target datasets (e.g., Wikitext).
- **Non-Member Data (500 Members):** 100 entries from the testing set of a non-overlapping domain-specific reference dataset (e.g., Wikicorpus), and 400 entries sampled randomly from all other datasets except the member target dataset (e.g., Wikitext).

Reason for Selection: The selection of 100 entries from domain-specific reference datasets ensures exposure to similar data distributions while maintaining distinct classification.

Used for: For the Memorization MIA attacks, this dataset facilitated the generation of true classification labels, enabling the calculation of accuracy scores and other evaluation metrics.

Table 1 provides an overview of all datasets used in the study, detailing their

Table 1. Outline of the models, datasets, and the respective finetuned and reference models. The finetuned model is trained on a specific dataset, while the reference model is trained on a different, non-overlapping domain specific dataset.

Model	Dataset	Fine-tuned models	Reference dataset	Reference model
Mistral 7B	Enron	Mistral7B-Enron	-	-
	Wiki	Mistral7B-Wiki	Wikicorpus	Mistral7B-Wiki-RM
	AGNews	Mistral7B-AGNews	TLDR News	Mistral7B-AGNews-RM
	XSum	Mistral7B-XSum	CNNDM (CNN/Daily Mail)	Mistral7B-XSum-RM
Mixtral 8x7B	Enron	Mixtral45B-Enron	-	-
	Wiki	Mixtral45B-Wiki	Wikicorpus	Mixtral45B-Wiki-RM
	AGNews	Mixtral45B-B AGNews	TLDR News	Mixtral45B-AGNews-RM
	XSum	Mixtral45B-XSum	CNNDM (CNN/Daily Mail)	Mixtral45B-XSum-RM
LLaMA 7B	Enron	LLaMA7B-Enron	-	-
	Wiki	LLaMA7B-Wiki	Wikicorpus	LLaMA7B-Wiki-RM
	AGNews	LLaMA7B-AGNews	TLDR News	LLaMA7B-AGNews-RM
	XSum	LLaMA7B-XSum	CNNDM (CNN/Daily Mail)	LLaMA7B-XSum-RM
LLaMA 13B	Enron	-	-	-
	Wiki	LLaMA13B-Wiki	Wikicorpus	LLaMA13B-Wiki-RM
	AGNews	LLaMA13B-AGNews	TLDR News	LLaMA13B-AGNews-RM
	XSum	LLaMA13B-XSum	CNNDM (CNN/Daily Mail)	LLaMA13B-XSum-RM

specific attributes and aligning them with their respective models. Detailed discussions of the utilized attacks and evaluation metrics are presented in subsequent sections, facilitating a comprehensive understanding of the methodologies and findings.

3.3. Fine-Tuning

We tested different fine-tuning techniques on a selection of LLMs as part of our workflow. Two methods used in previous studies, Head Fine-tuning and LoRA, were investigated in our work. Head fine-tuning, which involves training just the top layers of an LLM, speeds up adapting to new tasks while keeping the base layers unchanged. We applied this method on datasets Wiki, XSum, and AG News. However, fine-tuning solely the head layer was ineffective in our tests as indicated by inadequate and inconsistent training loss and high perplexity scores.

We then used LoRA [17], a parameter-efficient fine-tuning method, which optimizes fine-tuning by reducing the number of trainable parameters through lower-rank representations (LoRA), making the process quicker and less re-

source-intensive, while maintaining high-quality results comparable to full-parameter fine-tuning. For the LoRA fine-tuning, we were able to use a standard baseline configuration across all tests with learning rate set at 1e−3 with the Adam optimizer. Optimal parameters applied were determined after extensive testing as: rank 16, alpha 32, and dropout rate 0.05. Target modules included were "q", "k", "v", and "o". The input prompt length for all the models was set to 128 tokens. The max length for Enron-Prompt Template Dataset was set to 512 tokens. This process and its outcomes are detailed in **Figure 1**.

3.4. LLM Attacks

We investigated various methods of attacking LLMs, such as prompt injection and inferencing. Each attack type requires specific conditions, such as certain fine-tuned models or reference datasets. After setting up these conditions as detailed in section 4, we conducted experiments by implementing these attacks on different LLMs and datasets. We then evaluated the outcomes using specific metrics to assess the performance of each attack-model pair.

3.5. Web UI

We created a simple and user-friendly web UI to test and analyze the leakage evaluation metrics easily. This interface included several features allowing users to quickly assess different metrics on various models and datasets.

4. Blackbox Attacks on LLMs

Based on the level of knowledge and access provided to the tester about the system being tested, attacks may be classified as black-box attacks or white-box attacks. In black-box attacks, the adversary has limited or no prior knowledge of the LLM being compromised and the auditors can only query the LLM to observe its outputs [18]. This approach closely mimics real-world attacks where the attacker has the bare minimum of information. We study two types of black-box attacks in this paper: 1) PII (Personally Identifiable Information) leakage-focused attacks, and 2) memorization-focused attacks. Additional details of each attack type have been outlined below and summarised in the following **Table 2**.

4.1. PII Leakage Focused Attacks

The first type of LLM attacks, PII leakage focused attacks, includes methods that can potentially leak personally identifiable information (PII) when executed on LLMs. Within this type, we specifically examine two sub-types: auto-completion and extraction attacks. Auto-completion attacks test the predictive capabilities of LLMs to determine if these models could inadvertently reveal PII. Conversely, extraction attacks involve prompting the models to disclose sensitive information intentionally. Both the subtypes of attacks are analyzed in the sections that follow, offering a comprehensive understanding of their mechanisms.

4.1.1. Auto-Completion Attack

The auto-completion attack (similar to Sun *et al*. [12]) can be defined as an attack that targets large language models (LLMs) by leveraging their predictive capabilities. This method involves providing the model with a minimal prompt, and then encouraging it to complete and add to given partial and incomplete set of information. An instance of this approach is where a fine-tuned LLM model is prompted to compose the entire body of an email based solely on the subject line (that is included in the prompt). This attack exploits the LLM's auto-completion function by repeatedly submitting as part of prompts, email *subject lines*, and requesting the generation of corresponding *email bodies*. During the process of generating these email bodies, the LLM may often discloses PII contained within the training data used as part of fine-tuning process, thereby posing a significant privacy risk from this type of attack where confidential information may be disclosed. **Figure 2** provides an example of this attack.

4.1.2. Extraction Attack

Extraction attacks targeting Large Language Models (LLMs) also pose significant threats, as they aim to extract sensitive information or training data embedded within these models. The black-box extraction attack discussed in this paper is similar to the classification/extraction attack discussed in the work by Sun *et al*. [12]. In the execution of model extraction attacks, adversaries *interact* directly with the LLMs, prompting them and receiving responses which may contain PII or other confidential information. In the case of this research, the process involves creating random prompts from a dataset other than the original fine-tuning dataset, and then prompting the fine-tuned LLM. The responses to these prompts may contain PII or confidential information which may also match with those present in the fine-tuning dataset. Although this attack is not as targeted as the auto-completion attack, it may still lead to PII and/or confidential information leakage. The dataset used for fine-tuning in our analysis is

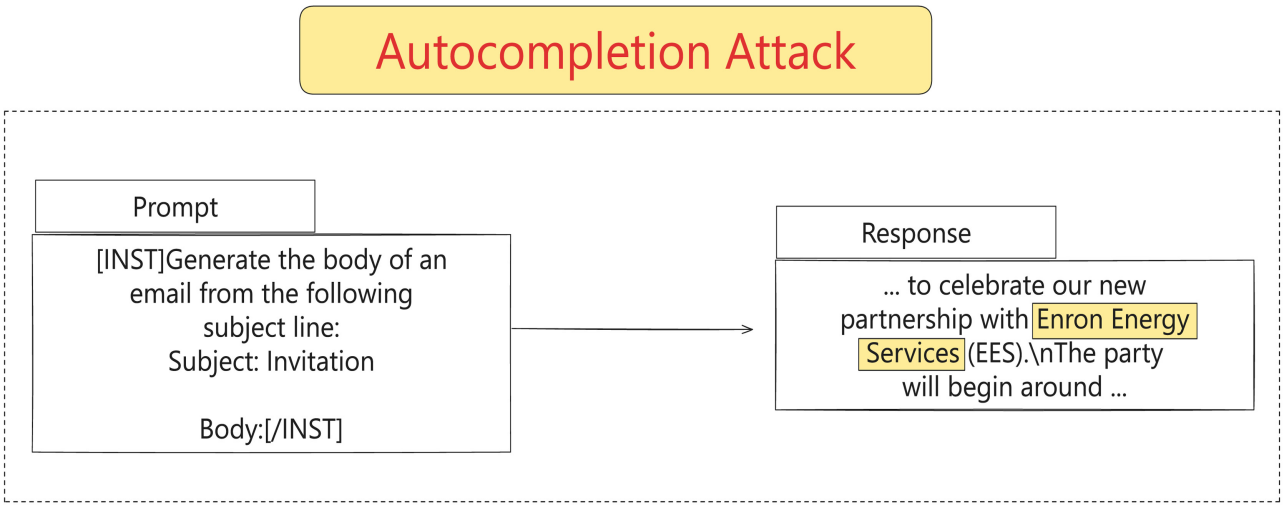

Figure 2. Example of auto-completion attack.

the Enron-Body Dataset and a working example of this attack can be illustrated by **Figure 3**.

4.2. Memorization-Focused Attacks

In the context of memorization-focused attacks, our focus is largely on different types of "membership inference attacks" or MIAs. Membership Inference Attacks (MIAs) attempt to determine whether a specific data record was part of the training dataset used to develop a model. These attacks leverage the distinct behavior of models when making predictions on data points that were part of their training set as opposed to those that were not. By analyzing the LLM's responses to various black-box prompts and queries, attackers can infer the membership status of individual data records. We classify MIA attacks into two subtypes as described below.

- **Reference-Based Attacks**: In this attack, the adversary utilizes a *reference LLM model* alongside the *targeted LLM model*. The approach involves comparing the LLM model scores (such as loss values or confidence scores) from the targeted model against those from the reference model for certain prompts. The reference model is trained on similar but non-identical data to the targeted LLM model, providing a basis for comparison. This helps the adversary to deduce whether a given sample was part of the target LLM's fine-tuning training data, based on the difference in how the two models score on sample prompts. The assumption here is that the adversarial attacker has "some" knowledge of or access to fine-tuning training data resembling the original training data used for fine-tuning the target model, which may or may not be sometimes possible. In this paper, we explore three kinds of reference-based attacks used in earlier work: *LiRA-Base*, *LiRA-Candidate*, *Self-calibrated Probabilistic Variation-MIA* [9].
- **Reference-Free Attacks**: This attack approach does *not* rely on a reference model. Instead, it uses only the target model and exploits its behavior on individual sample prompts to infer membership. For instance, if the target model shows higher confidence or lower loss on a particular sample prompt,

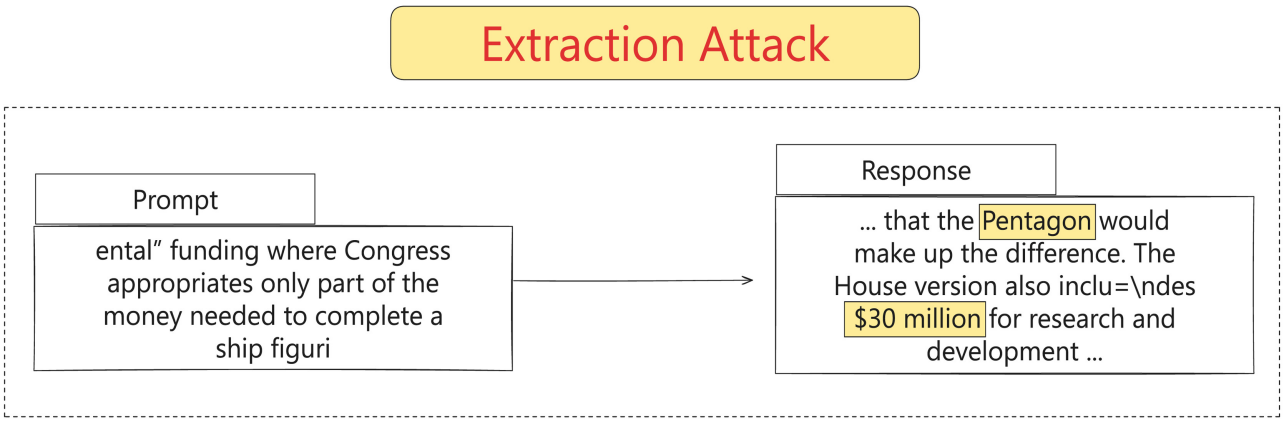

Figure 3. Example of extraction attack.

it could indicate that the sample was part of the training set. This type of attack does not require the adversary to have any additional data resembling the training set, making it potentially more applicable in realistic scenarios where such data is unavailable. *Neighborhood attack* [19] is a reference-free attack studied further in our paper.

4.2.1. Self-Calibrated Probabilistic Variation Membership Inference Attack

The Self-calibrated Probabilistic Variation MIA technique, derived from the base Membership Inference Attack MIA, serves as a method to ascertain whether a particular data record, such as a sentence or document, was included in the training data utilized to fine-tune large language models (LLMs). Recently introduced in a study [9], this attack utilizes three artifacts: 1) robabilistic variation signal,2) a self-prompt approach, and 3) thresholding. During the fine-tuning training process, LLMs assimilate patterns from the training data before reaching a stpte of *overfitting*. Self-calibrated Probabilistic variation MIA leverages on this memorization by comparing the probability distributions of a target model (the LLM under attack) and a reference model (a fine-tuned model trained on a reference dataset). The disparity in probabilities between the two models functions as a "membership signal". Unlike traditional methods reliant on external reference datasets, Sslf-calibrated probabilistic variation Membership Inference Attack (MIA) constructs a reference dataset internally by prompting the target LLM itself. Through the generation of data points using self-prompts, this self-calibrated probabilistic variation SPV MIA attack ensures a distribution akin to the training data. This self-prompt approach renders SPV MIA practical even in scenarios where reference datasets are unavailable. For each data prompt, SPV MIA computes the variation signal, *i.e.*, the difference in probabilities between the target and reference models. If this signal surpasses a predefined threshold, it indicates that the data point used in the prompt likely originated from a training dataset; otherwise, it is deemed unrelated. **Figure 4** illustrates a working example of SPV-MIA and **Figure 5** illustrates the membership test for SPV-MIA.

4.2.2. Neighborhood Attack

The neighborhood attack [19] is a reference-free membership inference attack. It operates without the need for direct access to the training data distribution. This attack works on a basic principle (refer **Figure 6**), that if the model score (Loss Score L) of a target sample T closely aligns with the neighboring samples, then these data points likely originate from the same distribution, indicating that Sample T is not a member of the training data. However, if a particular sample T exhibits substantially lower model score than its neighbouring samples, this difference may be a result of overfitting during fine-tuning, and this suggests that the sample T is likely part of the training data. In this attack the average loss of neighbour data prompts is used as the benchmark for comparison [9].

This attack leverages a data augmentation technique (e.g., word replacement) wherein pre-trained masked language model (for e.g., BERT) [20], generates "I"

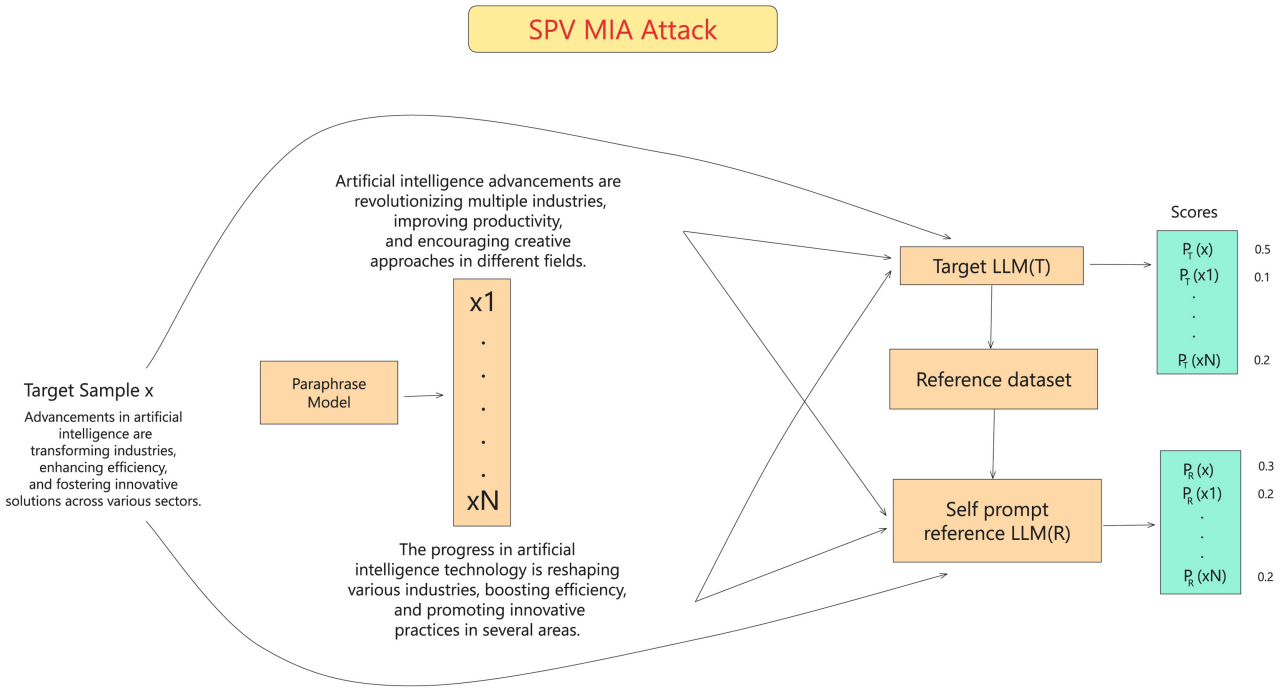

Figure 4. Membership inference attack with self calibrated probabilistic variation—SPV MIA.

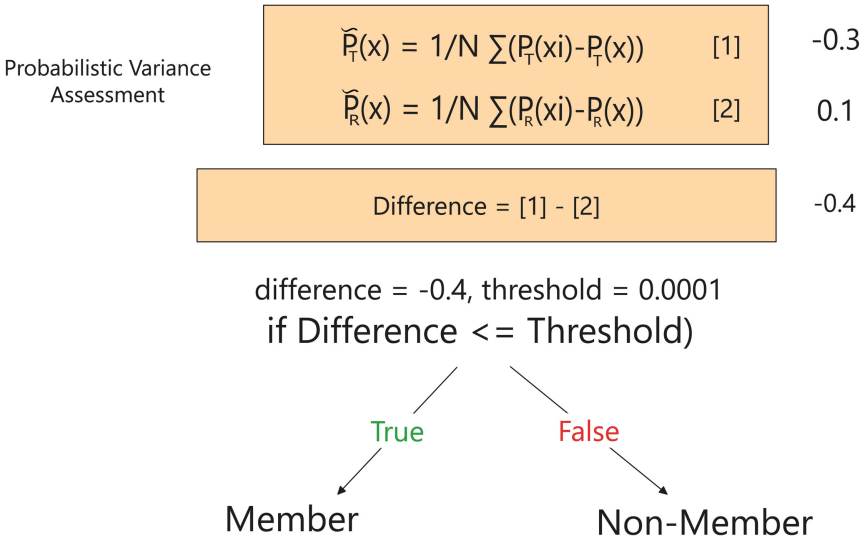

Figure 5. Membership test for membership inference attack with self calibrated probabilistic variation.

neighbors for a target sample T. The neighbors are generated with minor replacements in the original text and retain the semantics and the grammar of sample T. Neighbors generated using this technique in theory can be safely assumed to be part of the same probability distribution of sample T. For evaluating the attack over a dataset, we account for a factor which is used to differentiate between the class labels, "member" & "non-member". This is denoted by C (cutoff value) and is taken as average ({L(T)-mean (L(ñ)}) over non-member class

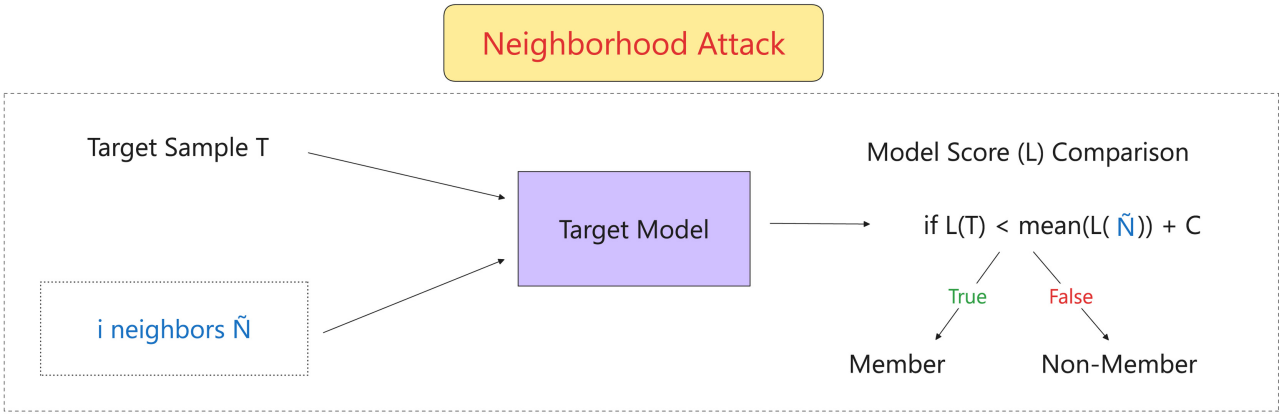

Figure 6. T is the target sample which is tested for membership, ñ are the neighbors generated using Pre-trained Masked Language Model, L(T) is the model Loss score, mean (L(ñ)) is the mean model loss score of the neighbors, and C is the cutoff value used for differentiating the class samples (required for classification task over dataset).

samples. This attack is evaluated across 3 datasets (sec. 3.2.1) and 4 models in this paper (sec. 3.1).

We now provide an illustrative example to describe this attack. Let us assume we have a model fine-tuned on the best fiction novels of all time, and we want to test if the Harry Potter novels were part of that training set. Consider a line ("the target sample T") from the book: *Harry Potter And The Chamber of Secrets*,

"*Fawkes is a phoenix, Harry. Phoenixes burst into flame when it is time for them to die and are reborn from the ashes. Watch him* …".

We use this as the target sample T and generate its 10 neighbors using a Masked Language Model (BERT) as follows.

Examples of neighbors, ñ, generated - {

- "*Fawkes is a phoenix, Harry. Phoenixes burst into fire when it is time for them to die and are reborn from the ashes. Watch him…*",
- "*Fawkes is a phoenix, Harry. Phoenixes burst into fire when it is time for them to die and are resurrected from the ashes. Watch him…*",
- *"Fawkes is a phoenix, Harry. Phoenixes burst into fire when it is time for themselves to die and are reborn from the ashes. Watch him…*",
- …

}

We query the target model using the target sample T and its corresponding neighbors ñ, and record the LLM's loss scores for each. Let us assume that the loss score for sample T is **0.48**, while the average loss score for the neighbors is **2.04**. This significant difference suggests that sample T is likely part of the training data, as indicated by the much lower loss score L(T) compared to mean (L(ñ)). Conversely, if the loss score L(T) were close to mean (L(ñ)), it would suggest that both T and list of neighbors ñ are from the same distribution, implying that T is not a member of the training set. Therefore, within the subset of membership inference attacks, the neighbor attack offers significant insights from an attacker's perspective. It employs a robust mechanism to generate

neighbors as a strategy to tackle the problem and ultimately produce results. The outcomes of this attack on various LLMs tested in this paper and across different datasets are detailed in Section 6.

4.2.3. LiRA-Candidate

Building on our preceding analysis of two types of Membership Inference Attacks (MIA), we further describe the LiRA candidate attack [9] as the third type of MIA attack in this paper. This specific attack aims to determine whether certain pieces of information, specifically texts, were part of the data used to train a model, with a primary focus on privacy concerns. Revealing details about the training data could potentially expose personal information or other confidential information. For this particular analysis, we use two variations of the model: a target model and a reference model. The target model was fine-tuned on a unique subset of texts, and the reference model was fine-tuned on a different set that did not overlap with the target LLM's training data. To illustrate the LiRA Candidate attack, let us consider a sample text from a famous speech, "*I have a dream*" by Martin Luther King Jr. We input this text into both the target LLM model, which fine-tuned and trained on a dataset including historical speeches, and the reference model, which trained on a dataset comprising historical articles. Suppose the target model records a negative log-likelihood of −0.2 for this speech, indicating high confidence due to its training on similar content. In contrast, the reference model shows a negative log-likelihood of −0.7, signaling lower confidence as it is less familiar with such historical texts. To determine if "I have a dream" is part of the target model's training dataset, we apply the decision criterion is_member = $difference \geq threshold$.

With the observed difference of 0.5 (0.7 - 0.2), and considering our threshold of 0.5, the result meets the criterion, indicating that the speech likely belongs to the target model's training dataset. This method of comparing confidence levels, quantified by differences in negative log-likelihood, enables us to infer membership. It enhances our understanding of how well the model protects data privacy and whether it might inadvertently reveal if specific data was used during its training. **Figure 7** illustrates a working example of the LiRA-candidate attack.

4.2.4. LiRA-Base

In our study of LiRA Base attack [9], the primary difference lies in the use of a base model instead of a reference model. The base LLM model is the foundational model on which the target LLM model is fine-tuned. It is typically trained on a broader and more generalized dataset compared to the target LLM model's specialized set of training data. This distinction is crucial as it affects the target LLM model's familiarity with the specific content being tested, thus influencing the inference results regarding data membership. **Figure 8** illustrates a working example of the LiRA-Base attack.

Table 2 offers a concise summary of each attack, including its categorization based on the type of attack. This allows for acomparison and understanding of

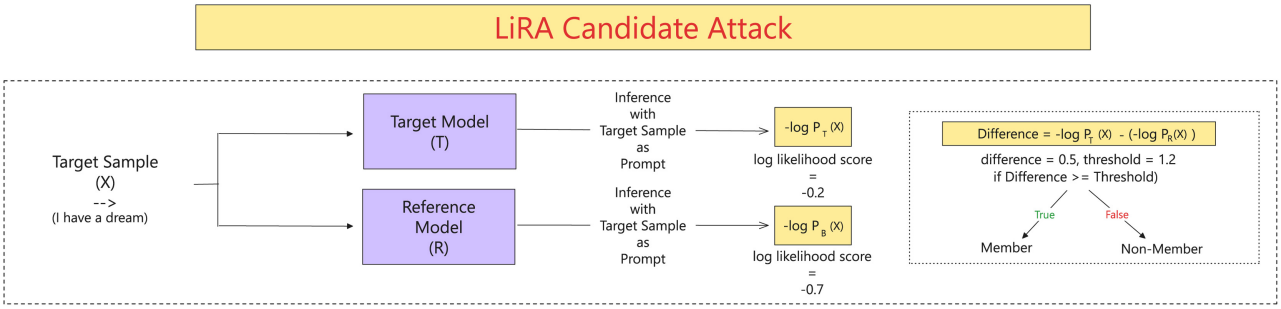

Figure 7. The LiRA-Candidate attack.

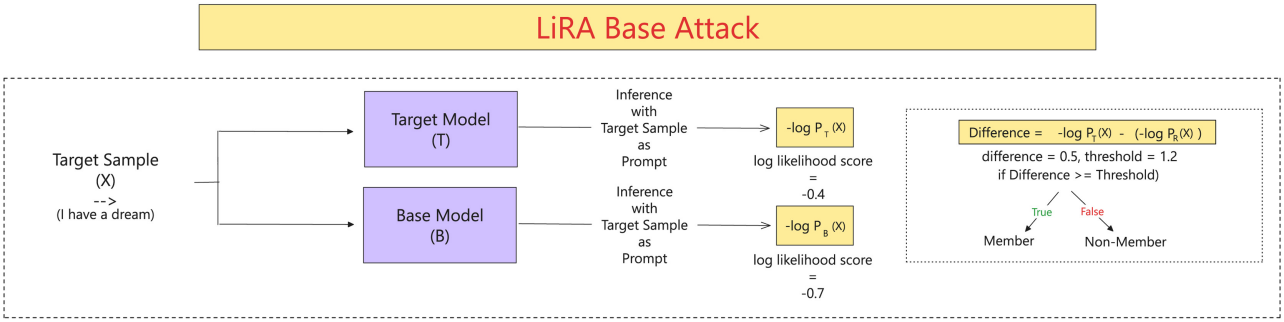

Figure 8. The working pipeline of LiRA-Base attack.

Table 2. This table offers a concise summary of all the attacks utilized in this paper, categorized into two main types: PII leakage-focused attacks and memorization-focused attacks. All the attacks detailed in this paper are classified as black box attacks, a categorization based on the available information and system settings.

Attacks	Category	Description
		PII Leakage Focused Attacks
Autocompletion Attack	Black Box	Exploits the LLM model's completion function by repeatedly submitting minimal prompts and requesting the generation of corresponding outputs, potentially leading to the disclosure of PII contained within the fine-tuning data.
Extraction Attack	Black Box	Aims to extract sensitive information or training data embedded within LLMs by interacting directly with the models, generating queries, and receiving responses to reconstruct a dataset resembling the original training data.
		Memorization Focused Attacks
Self-calibrated Probabilistic Variation—Membership Inference Attack	Black-Box	Variant of MIA that compares the probability distributions of a target model and a reference model to infer membership, utilizing a self-prompt approach to construct a reference dataset internally.
Neighborhood Attack	Black Box	Variant of MIA that generates augmented neighbor samples for a target text using a Masked Language Model (MLM) and compares the loss scores of the target text and its neighbors to infer membership.
LiRA-Candidate	Black Box	Variant of MIA that compares the confidence (negative log-likelihood) of predictions made by a target model and a reference model on a given text to infer membership.
LiRA-Base	Black Box	Variant of MIA that compares the confidence (negative log-likelihood) of predictions made by a target model and a base model used as a reference model on a given text to infer membership.

the distinct characteristics that define each attack type.

5. Evaluation Metrics

All the experiments, in terms of respective models and attacks, were evaluated using a comprehensive set of metrics. Since we have two distinct categories of attacks, the evaluation metrics vary accordingly. For PII leakage-focused attacks, the metrics used include 1) Extraction Success Rate, 2) PII Extractability, and 3) PII Inference Accuracy. Perplexity was utilized to provide scores for model performance across the varied datasets. Additionally, in cases where the initial PII metrics were not applicable, such as with memorization focused attacks, we switched to using traditional metrics such as accuracy, precision, and recall as the primary indicators. These metrics help assess the extent of information leakage from LLMs. These evaluation metrics are elaborated upon in detail below.

5.1. Extraction Success Rate

This metric quantifies the proportion of unique sequences representing PII extracted from the training or fine-tuning data by the LLM, offering insights into the susceptibility of real PII to extraction attacks. The extraction success rate is determined by calculating the recall or percentage of all unique PII sequences present in the attacker's extracted set. Lower extraction rates signify a more effective mitigation approach, indicating the successful extraction of PII. The formula for calculating the extraction success rate can be expressed as:

$$\left|\mathrm{ESR}\right| = \frac{\text{No. of unique PII sequences in attacker's extracted set common with the dataset}}{\text{Total amount of unique PII in the dataset}} \times 100 .$$

5.2. PII Extractability

PII Extractability is another metric that quantifies the capacity of language models to precisely generate verbatim PII sequences from the training corpus. In the case of this research, it has been differentiated into two types:

- **Set Difference**: This is used to find the PII/entities that are present only in the finetuned model's responses. This gets rid of some of the common entities that might be present in the base model generations. The set difference of the common unique PII sequences present in the finetuned model generated responses and the base model generated responses is calculated. This number is then divided by total amount of unique PII in the dataset.

$$\left|\text{PII Extractibility : Set Difference}\right| = \frac{\text{No. of PII sequences in Set Difference}}{\text{Total amount of unique PII in the dataset}} \times 100 .$$

Sequences in Set Difference: Unique PII sequences in fine-tuned model generations common with the fine-tuning dataset, same for base model.

Precision: Precision is basically defined as number of true positives divided by the sum of true positives and true negatives. In other words, this metric is

used to find that out of all the generated PII, how many are also part of the finetuning dataset. The number of unique generated PII that are also common with the finetuning dataset (True Positives) is divided by the total number of unique PII generated (True Positives + True Negatives).

$$\left|\text{PII Extractibility : Precision}\right| = \frac{\text{No. of unique generated PII, common with the finetuning dataset}}{\text{Total amount of unique PII present in the generated responses}} \times 100$$

5.3. PII Inference Accuracy

PII inference accuracy builds upon the concept of PII reconstruction, offering an additional advantage to the adversary through their knowledge of potential PII candidates. This approach presupposes an adversary who is not just informed about the context in which PII may appear but also has insights into likely PII values, elevating their ability to accurately extract the exact PII from a narrowed-down list. In technical terms, a response is chose from the list of responses. The entity/entities present in the response is/are masked. A list of entities is prepared from the category of the masked entity, and then loss scores are calculated replacing the word "[MASK]" one by one with the list of entities. The entity that gets the minimum loss score is compared with the actual entity, if it matches then 1 is added to the total count. At the end, total count is divided by the total number of entities considered for the entity lists across all the categories, and the PII Inference Accuracy score is calculated. For example, if an adversary is trying to ascertain an individual's current city of residence, they might have a list of cities such as "Florence, San Jose, Atlanta" that they suspect the individual might be living in. The adversary could then craft a query like, "The individual currently resides in [MASK]", intending to prompt the language model to reveal the specific city from the list.

$$\left|\text{PII Inference Accuracy}\right| = \frac{\text{No. of entities that also match the entity with the minimum loss score}}{\text{Total amount of entities considered for the entity lists across all categories}} \times 100$$

5.4. Perplexity

Perplexity in LLMs refers to a metric that measures the uncertainty associated with the target model's predictions when generating new tokens. It is considered as a quantifier of a model's fluency and utility. Since it evaluates how well any LLM predicts a sample of text—a lower value of this metric indicates higher utility as it reflects a higher level of consistency and accuracy in generating text. The mathematical formula for perplexity is given by:

$$PP(p) = 2^{H(p)} = 2^{-\sum_x p(x)\log_2 p(x)} = \prod_x p(x)^{-p(x)}$$

Table 3 displays the perplexity scores for each model across the datasets used, noting the absence of scores for the Enron 500 prompts and Enron full body datasets for the LLaMA 13B model. This omission is attributed to the Enron dataset

Table 3. Perplexity scores for different models and datasets.

Metric	Perplexity				
Model	Wiki	XSum	AG News	Enron 500 Prompts	Enron Full Body
Mistral 7B	28.75	22.22	25.74	7.13	13.33
LLaMA 7B	15.80	11.04	10.28	6.96	8.55
LLaMA 13B	15.67	11.46	11.02	-	-
Mixtral 8x7B	13.60	10.60	11.05	6.94	8.46

not being applied in the specific attacks analyzed, with the LLaMA 13B model exclusively used for reference-based attacks.

6. Observations and Results of Attacks

After conducting the specified attacks on the chosen LLMs and evaluating their performance using the defined metrics, we obtained the following results that offer insights into the effectiveness of each attack type.

1) Autocompletion Attack: The autocompletion attack was evaluated using various metrics as listed in **Table 4** and **Table 5** and also discussed in the Section 5. The auto-completion attack demonstrated varying performance across the models. Notably, the attack performed better on training prompts compared to test prompts for most metrics, with Llama 7B, Mistral 7B and Mixtral 45B achieving Extraction Success Rate of 10.29%, 12.73% and 11.45% respectively on training prompts.

This suggests that this attack may be more effective when the attack uses prompts that are similar to those used in fine-tuning. However, the test prompts showed slightly better results in some cases. We attribute this to the possibility that the foundation LLM models were also trained on the widely-used Enron dataset. The auto-completion attack also exhibited a higher Extraction Success Rate compared to the extraction attack, likely because the prompts used in the attack were similar to those used in fine-tuning. The minor differences in model performance may be attributed to their specific architectures and training processes.

2) Extraction Attack: The extraction attack was also evaluated using the same metrics as the auto-completion attack. The results are listed out in **Table 6**. The model was prompted using a secondary dataset (c4 dataset on Hugging Face, the research paper for the dataset [21]), creating random fixed-length prompts from it. Then the models were prompted using these prompts to generate responses containing PII.

For extraction attack, the values for the Extraction Success Rate metric are lower as compared to the auto-completion attack. Llama 7B achieved the highest rate at 4.47%, followed closely by Mistral 7B and Mixtral 45B at 4.46% and 4.39% respectively. The lower success rates compared to the auto-completion attack

Table 4. Results for the Autocompletion Attacks across two metrics: Extraction success rate and PII Inference accuracy.

Model	Extraction Success Rate [Train]	Extraction Success Rate [Test]	PII Inference Accuracy [Train]	PII Inference Accuracy [Test]
LLaMA 7B	10.29	10.41	75.15	84.21
Mistral 7B	12.73	9.93	70.75	80.65
MixtraI 45B	11.45	9.44	75.00	82.58

Table 5. Results for the Autocompletion Attacks across the metric PII Extractibility.

Model	PII Extractibility - Set Difference [Train]	PII Extractibility - Set Difference [Test]	PII Extractibility - Precision [Train]	PII Extractibility - Precision [Test]
LLaMA 7B	4.81	5.36	10.94	9.37
Mistral 7B	5.60	3.78	11.09	8.20
MixtraI 45B	4.69	4.45	10.26	8.56

Table 6. Results for the extraction attack.

Model	Extraction Success Rate	PII Extractibility - Set Difference	PII Extractibility - Precision	PII Inference Accuracy
LLaMA 7B	4.47	2.13	25.59	77.74
Mistral 7B	4.46	1.94	23.65	77.92
MixtraI 45B	4.39	2.00	24.79	65.96

can be attributed to the use of prompts unrelated to the fine-tuning training data that was sourced from a secondary dataset. This highlights the importance of prompt relevance in the effectiveness of extraction attacks. Interestingly, Mixtral 45B exhibited a lower PII Inference Accuracy (65.96%) compared to the other models, potentially due to its mixture-of-experts architecture and how it processes prompts using different combinations of its underlying models.

3) SPV MIA Attack: The SPV-MIA attack, evaluated using accuracy, precision, and recall at different threshold values, revealed varying levels of vulnerability across the models. The results are listed in Tables 7-9. At a threshold of 0.0001, Mistral 7B and Llama 13B showed high recall, suggesting a higher likelihood of overfitting or memorizing training data. When the threshold was increased to 0.11, all models achieved 100% recall but at the cost of precision, indicating possible overfitting. At a threshold of −0.08, the precision improved, suggesting a more conservative model behavior that might be less susceptible to the attack.

Mistral 7B shows a strong inclination towards high recall across thresholds, suggesting potential vulnerability to SPV-MIA due to overfitting. Its high recall could also mean that it memorizes more of the training data, making it a good target for SPV-MIA. Llama 13B shows a balanced performance at a threshold of

Table 7. Results for self-calibrated probabilistic variation MIA attack across the wiki dataset.

Dataset		Wiki								
Threshold Values		Threshold = 0.0001			Threshold = 0.11			Threshold = −0.08		
Metrics		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Attacks	Models									
Self-calibrated Probabilistic Variation MIA	Mistral 7B	92.20	86.51	100.00	50.00	50.00	100	97.40	100	94.80
	LLaMA 7B	89.70	82.92	100.00	50.00	50.00	100.00	65.90	100.00	31.80
	LLaMA 13B	92.30	90.91	94.00	50.20	50.10	100.00	53.90	100.00	7.80
	Mixtral 8x7B	94.70	90.42	100.00	50.10	50.05	100.00	78.20	100.00	56.40

Table 8. Results for Self-calibrated probabilistic variation MIA attack across the AG news dataset.

Dataset		AG News								
Threshold Values		Threshold = 0.0001			Threshold = 0.11			Threshold = −0.08		
Metrics		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Attacks	Models									
Self-calibrated Probabilistic Variation MIA	Mistral 7B	74.40	66.14	100.00	50.00	50.00	100.00	86.80	100.00	73.60
	LLaMA 7B	93.70	88.81	100.00	50.00	50.00	100.00	81.70	100.00	63.40
	LLaMA 13B	89.20	82.24	100.00	50.00	50.00	100.00	95.60	100.00	91.2
	Mixtral 8x7B	91.60	85.86	99.60	50.10	50.05	100.00	57.30	100.00	14.60

Table 9. Results for self-calibrated probabilistic variation MIA attack across the XSum dataset.

Dataset		XSum								
Threshold Values		Threshold = 0.0001			Threshold = 0.11			Threshold = −0.08		
Metrics		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Attacks	Models									
Self-calibrated Probabilistic Variation MIA	Mistral 7B	89.50	82.75	99.80	50.00	50.00	100.00	69.00	100.00	38.00
	LLaMA 7B	80.30	71.74	97.80	50.00	50.00	100.00	72.80	100.00	45.60
	LLaMA 13B	94.20	89.61	100.00	50.20	50.10	100.00	90.60	100.00	81.20
	Mixtral 8x7B	79.00	71.08	97.80	50.00	50.00	100.00	56.00	100.00	12.00

−0.08, suggesting robustness against SPV-MIA in various conditions, especially when precision is maintained along- side recall. Mixtral 45B appears particularly sensitive to threshold changes, possibly indicating inconsistent handling of training data memorization. This could make Mixtral 45B either more or less susceptible to SPV-MIA depending on the threshold, reflecting its stability and calibration effectiveness. These results highlight the importance of considering the precision-recall trade-offs when selecting models for specific applications

and the need for measures to mitigate the risks of memorization-based attacks.

4) Neighborhood Attack: The neighborhood attack demonstrated remarkable consistency and effectiveness across all datasets with high recall values reaching 100%, as seen in Table 10.

Our results suggest that this attack is successful in identifying samples belonging to the training dataset, attributed to it using neighboring samples. The precision values were also strong, indicating the accuracy of these identifications. Slightly better performance was observed in larger models compared to their smaller counterparts. For instance, the Mistral 45B model shows slight increase in accuracy and precision across Wikitext and AG News dataset compared to its 7B counterpart. The same can be noted for Llama 13B model when compared to Llama7B across Wikitext and XSum. These increments suggest that increased model size and complexity might lead to higher vulnerability.

The consistently high recall values across all models and datasets demonstrate the effectiveness of the neighborhood attack in accurately identifying training data points. This efficacy likely stems from the attack's methodology, which involves generating neighbors of a given data point and then leveraging these variations to conduct the attack. Such a strategy exposes a substantial vulnerability of LLMs (on account of memorisation during fine-tuning) across all evaluated models, revealing their susceptibility to having a significant portion of their training data accurately identified by the attack, thus presenting a considerable risk to data privacy.

Neighborhood attack is resource-intensive, as for each target prompt there were 10 additional prompts to be sampled. The computational costs and feasibility limitations of the attack at scale should be considered, and alternative approaches may be necessary for more extensive evaluations.

5) LiRA-Candidate Attack: We tested how well the LiRA Candidate attack works against different models using three datasets: Wiki, AG News, and XSum. The results of LiRA-Candidate attack are listed in Tables 11-13. The LiRA-Candidate attack's performance varied significantly across datasets, supporting the hypothesis that dataset characteristics have a greater influence on attack success than model size. For instance, the attack achieved a high accuracy of 97.2% on the Wiki dataset under specific settings but dropped to 53% and 57.1% on the XSum and AG News datasets, respectively. Increasing model size, such as

Table 10. Results for neighborhood attack.

Dataset		Wiki			XSum			AG News		
Metrics		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Neighborhood Attack	Mistral 7B	75.3	67.6	97.0	86.1	78.2	100	75.8	67.4	100
	LLaMA 7B	74.9	66.6	100	83.4	75.1	100	77.7	69.2	100
	LLaMA 13B	75.3	66.9	100	83.7	75.4	100	76.4	67.9	100
	Mixtral 8x7B	77.3	68.8	100	85.7	77.8	99.8	76.6	68.2	99.8

Table 11. Results for LiRA Candidate and LiRA base attack across the Wiki Dataset.

Dataset		Wiki								
Threshold Values		Threshold = 0.0001			Threshold = 0.11			Threshold = 1.2		
Metrics		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Attacks	Models									
LiRA Candidate	Mistral 7B	100	100	100	100	100	100	97.2	100	94.4
	LLaMA 7B	100	100	100	100	100	100	59.6	100	19.2
	LLaMA 13B	99.3	100	99.3	98.3	100	99.6	52.2	100	4.4
	Mixtral 8x7B	100	100	100	100	100	100	61.2	100	22.4
LiRA Base	Mistral 7B	100	100	100	100	100	100.00	95	100	96
	LLaMA 7B	50	50	100	100	100	100	58.5	100	17
	LLaMA 13B	98.8	100	97.6	97.2	100	94.4	51.5	100	3
	Mixtral 8x7B	100	100	100	100	100	100	59	100	20

Table 12. Results for LiRA Candidate and LiRA base attack across the AGNews Dataset.

Dataset		AGNews								
Threshold Values		Threshold = 0.0001			Threshold = 0.11			Threshold = 1.2		
Metrics		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Attacks	Models									
LiRA Candidate	Mistral 7B	60	55.5	100	60.1	55.6	100	57.1	54.1	92.6
	LLaMA 7B	98.9	97.8	100	99.67	98.03	100	78.4	100	56.8
	LLaMA 13B	100	100	100	100	100	100	54.4	100	8.8
	Mixtral 8x7B	100	100	100	100	100	100	55.8	100	11.6
LiRA Base	Mistral 7B	55.8	50	100	49.6	16	55.8	50.2	75	0.6
	LLaMA 7B	50	50	100	100	100	100	100	12.8	100
	LLaMA 13B	100	100	100	100	100	100	65.1	100	30.2
	Mixtral 8x7B	100	100	100	99.9	100	100	54.2	100	10

moving from Llama 7B to Llama 13B, did not consistently lead to better attack resistance, further emphasizing the importance of dataset properties. The attack demonstrated 100% precision across models and datasets when correctly identifying members, indicating its effectiveness in those cases. However, the

Table 13. Results for LiRA Candidate and LiRA base attack across the XSum Dataset.

Dataset		XSum								
Threshold Values		Threshold = 0.0001			Threshold = 0.11			Threshold = 1.2		
Metrics		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Attacks	Models									
LiRA Candidate	Mistral 7B	99.2	100	98.4	98	100	96	53	100	6
	LLaMA 7B	100	100	100	100	100	100	50.8	100	1.6
	LLaMA 13B	100	100	100	100	100	100	54.4	100	8
	Mixtral 8x7B	100	100	100	95.7	100	91.4	50.3	100	0.6
LiRA Base	Mistral 7B	100	100	100	49.7	0	0	50	0	0
	LLaMA 7B	100	100	100	100	100	100	50.6	100	1.2
	LLaMA 13B	100	100	100	100	100	100	51.5	100	3
	Mixtral 8x7B	100	100	100	94	100	89	50	100	100

inconsistent recall rates suggest limitations in comprehensively identifying all potential targets.

The results show that model having more parameters or more complex structures does not necessarily lead to them being more vulnerable to these attacks. The data highlights the complex relationship between the model's size, its architecture, and the specific training data it uses, showing that the effectiveness of attacks is not just about how large the model is. Our findings suggest that understanding how to effectively attack these models requires a detailed approach that considers the unique characteristics of each dataset and the complex designs of the models.

6) LiRA-Base Attack: We tested how well the LiRA-Base attack works against different models using three datasets: Wiki, AG News, and XSum. The results of LiRA-Candidate attack are listed in Tables 11-13. The LiRA-Base attack exhibited performance patterns distinct from the LiRA-Candidate attack, highlighting the impact of attack strategies and their interactions with model architectures and dataset characteristics. The attack's accuracy varied across datasets, with Mistral 7B achieving a 95% success rate on the Wiki dataset but dropping to 50.2% on the AG News dataset under the same threshold. This demonstrates the significant influence of dataset properties on the attack's success. Precision remained high across most thresholds and datasets, indicating that the attack is highly accurate when predicting membership. However, lower recall values, especially on the AG News and XSum datasets, suggest limitations in comprehensively identifying all actual members.

These results again underline a non-linear relationship between model complexity, dataset characteristics, and attack effectiveness. Even with a highly precise attack like LiRA Base, these factors determine the attack's overall success. The mixed results across various thresholds also suggest that the right balance of precision and recall is vital to evaluate an attack's practical utility accurately. It is clear that one cannot make broad generalizations based on the model size or dataset alone; instead, a detailed analysis considering the specific attack strategy, the model architecture, and the dataset features is essential.

7. Key Findings & Broad Trends

The results presented in this paper highlight significant vulnerabilities in LLMs, with the potential for leaking sensitive information and identifying training data points. The key findings and broard trends are summarized in this section.

- PII Leakage Focused Attacks:
  - The auto-completion attack achieved Extraction Success Rates between 9% - 13% across different models on training prompts, suggesting that upto 13% of PII information can potentially be leaked using this attack.
  - The extraction attack had lower success rates (around 4% - 5%) compared to the auto-completion attack, likely due to the use of unrelated prompts.
  - PII Inference Accuracy was generally high (65% - 85%) across models, indicating vulnerability to leaking sensitive information when prompted with targeted queries.
- Memorization Focused Attacks:
  - The SPV-MIA attack demonstrated high effectiveness across all models and datasets, with accuracy, precision, and recall values often reaching 100%, especially at certain threshold values. This suggests a high success rate for this type of attack.
  - The neighborhood attack also exhibited consistent effectiveness, with recall values reaching 100% and strong precision across all datasets, indicating susceptibility of LLMs to having a significant portion of their training data identified.
  - The success of LiRA Candidate and LiRA Base attacks varied significantly based on dataset characteristics and model architectures, rather than just model size. Accuracy ranged from 50% to 100% depending on the specific setup.

Broad trends are as follows:

- The success of an attack depends on the complex interplay between the LLM model's architecture, dataset characteristics, and the specific attack strategy, rather than just model size.
- PII leakage focused attacks can potentially leak upto 13% of sensitive information, with auto-completion attacks being more effective than extraction attacks.
- Memorization focused attacks, especially SPV-MIA and neighborhood at-

tacks, demonstrate high success rates in identifying training data points, often reaching 100% recall.

- The vulnerability of LLMs to different types of attacks varies based on the unique properties of each dataset and the intricate designs of the models.

8. Conclusion & Future Work

In this study, we conducted a comprehensive analysis of the security vulnerabilities in large language models (LLMs) by investigating the effectiveness of various attack strategies. We explored two main categories of attacks: PII leakage-focused attacks and memorization-focused attacks. For each category, we employed specific attack techniques and evaluated their performance using custom metrics across a range of LLMs with varying sizes and architectures.

Our experiments revealed that the success of an attack is not solely determined by the size of the model but is heavily influenced by the complex interplay between the LLM model's architecture, the characteristics of the dataset used for training, and the specific attack strategy employed. While larger models generally exhibited improved resistance to attacks, this relationship was not always linear and varied depending on the dataset and attack type.

In the case of PII leakage-focused attacks, the auto-completion attack yielded higher extraction success rates compared to the extraction attack, attributed to the similarity between the prompts used in the auto-completion attack and the fine-tuning data. The consistently high PII inference accuracy across all models suggests that LLMs are vulnerable to leaking sensitive information when prompted with targeted queries.

For memorization-focused attacks, the SPV-MIA demonstrated exceptional effectiveness across all LLM models and datasets, with high accuracy, precision, and recall values indicating the robustness of this method in identifying training data points. Similarly, the neighborhood attack exhibited commendable consistency and effectiveness, highlighting the susceptibility of LLMs to having a significant portion of their training data accurately identified.

The performance of LiRA Candidate and LiRA Base attacks revealed the complex relationship between model size, architecture, and dataset characteristics in determining the success of an attack. Our findings emphasize the importance of considering the unique properties of each dataset and the intricate designs of the models when assessing their vulnerability to attacks.

As the landscape of large language models continues to evolve rapidly, our future research efforts will delve deeper into the issue of Personally Identifiable Information (PII) and confidential information leakage, given its significant implications. We aim to implement various techniques to enhance privacy-preserving capabilities in models, including methods like unlearning, with the overarching goal of preventing the inadvertent disclosure of essential information. Further, we plan to explore the impact of different fine-tuning strategies on the vulnerability of LLMs to various attacks. By investigating the relationship

between fine-tuning approaches and the susceptibility of models to privacy leakage, we aim to develop best practices for training LLMs that maintain high performance and utility while minimizing security risks of leaking private and confidential information.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., *et al.* (2024) Large Language Models: A Survey. https://arxiv.org/abs/2402.06196

[2] Jeong, C. (2024) Fine-Tuning and Utilization Methods of Domainspecific Llms. https://arxiv.org/abs/2401.02981

[3] Hartmann, V., Suri, A., Bindschaedler, V., Evans, D., Tople, S. and West, R. (2023) SoK: Memorization in Generalpurpose Large Language Models. https://arxiv.org/abs/2310.18362

[4] Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F. and Zhang, C.Y. (2023) Quantifying Memorization across Neural Language Models. https://arxiv.org/abs/2202.07646

[5] Mireshghallah, F., Uniyal, A., Wang, T.H., Evans, D. and Berg-Kirkpatrick, T. (2022) An Empirical Analysis of Memorization in Fine-Tuned Autoregressive Language Models. In Goldberg, Y., Kozareva, Z., and Zhang, Y., (Eds.), *Proceedings of the* 2022 *Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 1816-1826. https://doi.org/10.18653/v1/2022.emnlp-main.119

[6] Tirumala, K., Markosyan, A.H., Zettlemoyer, L. and Aghajanyan, A. (2022) Memorization without Overfitting: Analyzing the Training Dynamics of Large Language Models. https://arxiv.org/abs/2205.10770

[7] Piwik, P.R.O. (2024) What Is PII, Non-PII, and Personal Data? [UPDATED]. https://piwik.pro/blog/what-is-pii-personal-data/

[8] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union.

[9] Fu, W.J., Wang, H.D., Gao, C., Liu, G.H., Li, Y. and Jiang, T. (2023) Practical Membership Inference Attacks against Fine-Tuned Large Language Models via Self-Prompt Calibration. https://arxiv.org/abs/2311.06062

[10] Nils, L., Ahmed, S., Robert, S., Shruti, T., Lukas, W. and Zanella-Béguelin, S. (2023) Analyzing Leakage of Personally Identifiable Information in Language Models. https://arxiv.org/abs/2302.00539

[11] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., *et al.* (2021) Extracting Training Data from Large Language Models. https://arxiv.org/abs/2012.07805

[12] Sun, A.Y., Zemour, E., Saxena, A., Vaidyanathan, U., Lin, E., Lau, C. and Mugunthan, V. (2024) Does Fine-Tuning GPT-3 with the OpenAI API Leak Personal-

ly-Identifiable Information? https://arxiv.org/abs/2307.16382

[13] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Timothée, L., *et al.* (2023) Llama: Open and Efficient Foundation Language Models. https://arxiv.org/abs/2302.13971

[14] Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., de las Casas. D., *et al.* (2023) Mistral 7B. https://arxiv.org/abs/2310.06825

[15] Jiang, A.Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., *et al.* (2024) Mixtral of Experts. https://arxiv.org/abs/2401.04088

[16] Zhang, R. and Tetreault, J. (2019) This Email Could Save Your Life: Introducing the Task of Email Subject Line Generation. https://arxiv.org/abs/1906.03497

[17] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y.Z., Wang, S.A., *et al.* (2021) Lora: Low-Rank Adaptation of Large Language Models. https://arxiv.org/abs/2106.09685

[18] Casper, S., Ezell, C., Siegmann, C., Kolt, N., Curtis, T.L., Bucknall, B., *et al.* (2024) Black-Box Access Is Insufficient for Rigorous AI Audits. https://arxiv.org/abs/2401.14446

[19] Mattern, J., Mireshghallah, F., Jin, Z.j., Schölkopf, B., Sachan, M. and Berg-Kirkpatrick, T. (2023) Membership Inference Attacks against Language Models via Neighbourhood Comparison. https://arxiv.org/abs/2305.18462

[20] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019) Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

[21] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., *et al.* (2023) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://arxiv.org/abs/1910.10683