Nob-MIAs: Non-biased Membership Inference Attacks Assessment on Large Language Models with Ex-Post Dataset Construction

Cédric Eichler, Nathan Champeil, Nicolas Anciaux, Alexandra Bensamoun, Héber Hwang Arcolezi, Jose Maria de Fuentes

Nob-MIAs: Non-biased Membership Inference Attacks Assessment on Large Language Models with Ex-Post Dataset Construction

Cédric Eichler[1 2], Nathan Champeil[2 3], Nicolas Anciaux[2], Alexandra Bensamoun[4], Heber Hwang Arcolezi, and José Maria De Fuentes[5]

[1] Laboratoire d'Informatique Fondamentale d'Orléans, INSA Centre Val de Loire, Université d'Orléans, Bourges, France
[2] Inria, <firstname.lastname@inria.fr>
[3] ENSTA, Paris, France
[4] Université Paris-Saclay, alexandra.bensamoun@universite-paris-saclay.fr
[5] Universidad Carlos III de Madrid, <josemaria.defuentes@uc3m.es>

Abstract. The rise of Large Language Models (LLMs) has triggered legal and ethical concerns, especially regarding the unauthorized use of copyrighted materials in their training datasets. This has led to lawsuits against tech companies accused of using protected content without permission. Membership Inference Attacks (MIAs) aim to detect whether specific documents were used in a given LLM pretraining, but their effectiveness is undermined by biases such as time-shifts and n-gram overlaps. This paper addresses the evaluation of MIAs on LLMs with partially inferable training sets, under the ex-post hypothesis, which acknowledges inherent distributional biases between members and non-members datasets. We propose and validate algorithms to create "non-biased" and "non-classifiable" datasets for fairer MIA assessment. Experiments using the Gutenberg dataset on OpenLamma and Pythia show that neutralizing known biases alone is insufficient. Our methods produce non-biased ex-post datasets with AUC-ROC scores comparable to those previously obtained on genuinely random datasets, validating our approach. Globally, MIAs yield results close to random, with only one being effective on both random and our datasets, but its performance decreases when bias is removed.

Keywords: Membership Inference Attack · Assessment · Bias.

1 Introduction

The proliferation of Large Language Models (LLMs) has ignited significant legal and ethical debates, particularly concerning copyright infringement. These models often do not document their training data sources, leading to disputes over unauthorized use of copyrighted material. For instance, lawsuits are piling up against OpenAI accused of training ChatGPT using articles and other books

without permission [6]. Similar accusations have been leveled at Meta or Google for allegedly using protected content [7]. These issues underscore the societal, economic, and legal implications of LLM training practices.

Recent surveys highlight the challenges of respecting copyright in AI training across different jurisdictions, such as the USA and France. In the USA, the use of copyrighted data is generally prohibited without the rights holder's permission unless it falls under "fair use" [24]. For the culture and media sectors, LLM training could not be exempted from this limitation due to the conditions associated with it. More than twenty lawsuits are pending in the USA. In the European Union, the Directive 2019/790 on copyright and related rights in the digital single market introduces a "text and data mining" exception, for any purpose, that could correspond to the use of protected content for LLM training. However, its benefit is conditional on lawful access to copyrighted data and the absence of an opt-out by rights holders. However, not only do the training databases contain infringing works, but most of the rights holders have exercised their opt-out. However, the opacity of the process compromises the return to exclusive rights. So, to provide leverage, the AI Act (European Regulation 2024/1689), the first comprehensive regulation on AI, has required LLM providers, on the one hand, to put in place an internal policy aimed at respecting copyright and, on the other, to be transparent about the sources of training. The AI Office will provide a template on this point. The issue of knowledge of the use of content by the LLM is therefore crucial for rights holders.

Objective. From a technical perspective, determining whether a specific document was a member of the training set of a machine learning (ML) model based on the model's output, is a Membership Inference Attacks (MIA) problem, highlighted in 2017 by Shokri et al. [26]. However, the effectiveness of MIAs in the context of LLMs is subject to debate.

Limits of existing solutions. To determine whether a particular document has been used to train an AI model, MIA rely on overlearning, which results in stronger predictions when applied to training data. While some research shows that MIAs can achieve high accuracy [19,18], other studies [7,6] question their validity due to inherent biases in the datasets of members and non-members used for their assessment. For example, biases like time shifts and n-gram overlaps can lead to over-interpretation of results [7]. Additionally, studies indicate that some biases can be exploited to make a "blind" classifier, without model access, more effective than MIAs [6]. This raises doubts about the robustness and practical relevance of current MIA techniques, and recent surveys like [13] point out the lack of rigor and practical relevance of current proposals.

Research question and contributions. This paper aims to address the problem of assessing MIA effectiveness on LLMs which do not disclose their full training datasets but where part of the training dataset can be inferred. The research

[6] See, e.g., businessinsider.com, "The copyright lawsuits against OpenAI are piling up as the tech company seeks data to train its AI", Jun 30, 2024

[7] See, e.g., wired.com, "Congress Senate Tech Companies Pay AI Training Data", July 2, 2024

question we seek to answer is: How can we construct an unbiased dataset for evaluating MIAs on LLMs with partially inferable training sets?

We propose and evaluate two approaches: (1) Creating datasets that are "non-biased" by design with respect to known biases, and (2) Constructing datasets that cannot be classified, ensuring fairer assessment. Our contributions can be summarized as follows:

- We provide algorithms for constructing ex-post datasets of two types: $No-Ngram$ ("No N-gram bias") and $No{-}Class$ ("non classifiable"), each designed to mitigate specific types of biases for MIA assessment.
- We validate our algorithms and compare our proposed methods in the assessment of existing MIAs.
- We demonstrate that neutralizing known biases (e.g., time shifts, n-gram biases) is insufficient for accurate MIA assessment; we also show that several existing MIAs, which are presumed to be effective, are not efficient when evaluated using non-classifiable datasets. For instance, our experiments show that the TPR@10%FPR and ROC AUC of the best performing MIA out of the 6 assessed drop by 40% and 14.3% respectively when evaluated on datasets produced with our approach rather than on randomly sampled ones.

Outline. Section 2 reviews related work and positions our research. Section 3 synthesizes our hypothesis and the addressed problem. Section 4 presents our proposed solutions and algorithms, detailing ex-post (i.e., a posteriori) construction of unbiased datasets. Section 5 provides a comparative experimental evaluation of the proposed solutions using the Gutenberg dataset. Finally, Section 6 concludes the paper and suggests directions for future research.

2 Related Work and Positioning

Methods for adapting Membership Inference Attacks (MIAs), originally developed for machine learning classification algorithms [26], have recently been adapted to the context of Large Language Models (LLMs). Baselines technique use *likelihood-based* metrics such as Loss [31] and Perplexity [2] to distinguish between *members* (documents which were used for LLM training) and *non-members* (not used in training). Perplexity, noted here as π, is defined for a LLM $\mathcal{M}$ and a document D (e.g., a book) comprising a sequence $d_1, \ldots, d_{|D|}$ of tokens (words) as the average negative log-likelihood of these token sequences: $\pi(\mathcal{M}, D) = -\frac{1}{|D|}\sum_{i=1}^{|D|} \log(\mathbb{P}[\mathcal{M}(d_i|d_1, \ldots, d_{i-1})]$, where $|D|$ is the number of tokens in D, and $\mathbb{P}[d_2|d_1]$ denotes the conditional probability that token d_2 follows d_1. The MIA is performed using a threshold classification, considering D as a member of the training set if the π score is lower than γ: $MIA_\gamma(\mathcal{M}, D) = \mathbb{1}[\pi(\mathcal{M}, D) < \gamma]$. Variants include using the perplexity per bit of information after compressing D: $MIA_\gamma(\mathcal{M}, D) = \mathbb{1}[\pi(\mathcal{M}, D)/zlib(D) < \gamma]$, where $zlib(D)$ is the number of bits after compressing D using Zlib. Several studies show that likelihood-based MIAs applied to LLMs are effective, with

high AUC-ROC values. For example, [19] reports an AUC-ROC of 0.856 for books.

Other metrics, such as Min-k%Prob [25], are based on the premise that if the text has been read by the LLM, it appears with a higher probability. Min-k%Prob selects the k% tokens of the document with the minimum probabilities returned by the LLM and computes their average log-likelihood. Results show an AUC-ROC of 0.88 on ChatGPT for a dataset based on sentences from books, some of which are known to have been read by ChatGPT.

Many other variants of MIAs for LLMs have been proposed, also based on likelihood metrics. Neighboring-based MIAs, where the perplexity score is calibrated using either neighboring models $\mathcal{M}'$ or neighboring documents D'. Classification between members and non-members is then obtained by comparing the likelihood of the target model $\mathcal{M}$ and document D with that of the neighboring model $\mathcal{M}'$ and document D, or the model $\mathcal{M}$ and neighboring documents D'. Neighboring models hence assume access to a reference model trained on a disjoint data set drawn from a similar distribution, which is often unrealistic [7]. Neighboring documents [9] are more realistic but present slightly lower performance and additional difficulties in correctly setting noise parameters.

In cases where LLMs do not output likelihood information, complementary metrics can be used with acceptable performance penalty. For example, [14] proposes a MIA method called SaMIA, which measures the similarity between input samples from a document D and the rest of the text in D using ROUGE [16]. SaMIA demonstrates an AUC-ROC of 0.64 on subsets of The Pile dataset.

Recent studies [7,18] challenge the high-performance claims of MIAs on LLMs. They identify several biases that may skew results, such as timeshift between members and non-members, leading to different distributions of dates and word usage. Re-evaluating some MIAs on Pythia [1], trained on genuinely random train/test splits of The Pile [10] (and hence have no bias), shows decreased AUC-ROC measures, questioning the apparent success of some MIAs.

Some studies suggest that naive classifiers can distinguish members from non-members with good results based on these biases [6,20]. These studies conclude that MIAs must be evaluated on random datasets taken from a same distribution. While this is possible with open LLMs like Pythia, which reveal their training and test data sources, it is not feasible for LLMs that do not disclose their sources (those of interest in copyright cases). As shown in the literature (see, e.g., [7,18,21]), the same MIAs yield different performance (accuracy and relevance) results on different LLMs/datasets. Therefore, the assessment of MIAs on open LLMs cannot be directly transposed to LLMs that do not disclose their training dataset. This confirms the need for techniques like the one we propose.

A technique inspired by the Regression Discontinuity Design from causal inference, originally used to study treatment effects based on a cutoff date, is proposed in [20]. However, documents added just before or after the cutoff date must be known and sufficiently numerous. Additionally, declared dates often deviate from reality [4], making this approach impractical.

Many other works are based on MIA attacks on LLMs but rely on different hypotheses, leading to solutions not applicable to our context. [15] introduces a framework using loss gap variation during fine-tuning to detect if a document has been seen, though this is not generalizable to initial training documents and also requires an assesment using unbiased datasets. Related works on copyright aspects include copyright issues in LLM outputs [22,17] to reduce copyrighted text generation and protect users from potential plagiarism, and watermarking techniques [21,29] for detecting violations in LLM pretraining data, or LLM fine-tuning data [30], but these works are not transposable to our context.

3 Problem statement

A Membership Inference Attack MIA on a large language model $\mathcal{M}$ is a binary classification task aiming to determine whether a textual document D is included in the set of documents $\mathcal{D}_{\text{train}}$ used as the training dataset used to build $\mathcal{M}$. The goal of this attack is to design a function $MIA : \mathcal{D} \rightarrow \{0, 1\}$ that can ascertain the truth value of $D \in \mathcal{D}_{\text{train}}$ for any document in the document space $\mathcal{D}$.

In the context of copyright checks, our goal is to detect ex-post potential violations involving protected texts in the LLM's pretraining dataset. Our hypotheses H1 to H3 stem from this context, acknowledging that the LLM may try to obscure the use of these texts:

H1: Self-Assesment. We assume that a reliable assessment of the MIA must be performed on the target LLM $\mathcal{M}$ itself. As shown in the literature (see, e.g., [7,18]), the same MIAs yield different performance (accuracy and relevance) results on different LLMs/datasets.

H2: Partial Member Knowledge. The training dataset $\mathcal{D}_{\text{train}}$ of $\mathcal{M}$ is partially inferable, i.e., a subset $\mathcal{D}_{\text{train}}^{known}$ of $\mathcal{D}_{\text{train}}$ can be inferred by the attacker. For example, it is know that OpenAI models like GPT-4 have memorized some precise collections of copyright protected books [3].

H3: Bias Recognition. A subset $\mathcal{D}_{nm}^{known}$ of non-members (i.e., $\mathcal{D}_{nm}^{known} \subset \mathcal{D} \backslash \mathcal{D}_{\text{train}}$) is (obviously) known. Traditionally, such a subset is constructed by considering documents not available at the time the target LLM was released. This creates inherent biases in the ex-post context, where members and non-members are not drawn from the same ransdom distribution.

We address the problem of producing datasets of members $\mathcal{D}_{train}^{Nob} \subset \mathcal{D}_{train}^{known}$ and non-members $\mathcal{D}_{nm}^{Nob} \subset \mathcal{D}_{nm}^{known}$ which aim to minimize bias (hence the name "Nob" for Non-biased), ensuring a reliable evaluation of MIAs on LLMs while satisfying these three hypotheses.

4 Neutralizing Bias in Ex-Post Dataset Construction

In this section, we present our approach to identifying and mitigating specific bias in the construction of datsets used for the assesment of MIAs. Our methodology operates in two phases: first, addressing bias due to n-gram overlap, which

has been shown to significantly affect the assesment of MIAs [7], and second, addressing additionnal biases that go beyond n-gram overlap.

4.1 Methodology for Identifying and Mitigating Bias

We begin by targeting n-gram bias, as previous work has demonstrated that n-gram overlaps between members and non-members can distort MIA benchmarks [7]. To counteract this, we propose the $No - Ngram$ algorithm, which aims to generate members and non-members sets with similar distributions of n-gram overlaps.

Next, we leverage traditional classifiers, which we refer to as "LLM-Agnostic" classifiers, to identify and mitigate bias beyond n-gram overlap. These classifiers operate without any prior knowledge of the target language model $\mathcal{M}$ or the training dataset $\mathcal{D}_{train}$. Our approach uses these classifiers to create member and non-member datasets that resist effective classification. The $No - Class$ algorithm further neutralizes detectable biases, hindering the classifier's ability to distinguish between members and non-members.

4.2 Neutralizing N-gram Bias

The impact of n-gram distribution on MIA performance has been extensively documented. For example, time-shifted datasets often exhibit variations in n-gram distribution due to changes in dates, but also language, vocabulary and topics of interest over time [7]. A significant difference in n-gram overlap between non-members and left-out members can lead to an inflated evaluation of MIA performance. To mitigate this, we propose the $No - Ngram$ algorithm (see Algorithm 1), which generates member and non-member sets with distributions of n-gram overlap w.r.t. left-out members that closely match.

Algorithm overview. Algorithm 1 operates in three steps:

- *Initial sampling:* The algorithm begins by selecting an arbitrary sample $\mathcal{D}_{train}^{Nob}$ of $\mathcal{D}_{train}^{known}$ of an appropriate size (line 1).
- *Histogram computation:* It then computes the histogram of n-gram overlap between the selected members and remaining ones (line 3) using the function *histo* (described below). This histogram represents the target overlap distribution that the non-member set should mirror to be indistinguishable from selected members.
- *Greedy construction:* Afterwards, the non-member dataset is constructed document by document, in a greedy fashion: at each step (lines 5 to 9), the document that minimizes the Kolmogorov-Smirnov distance (see below) between the n-gram overlap histograms is selected.

The Kolmogorov-Smirnov (KS) distance[8] used in the algorithm is a widely used metric for measuring the distance between (real, non parametric) distributions. In the context of LLMs, it is particularly useful for comparing distributions

[8] Other distance metrics could be used. Exploring them is planned for future work.

Algorithm 1 $No - Ngram$

Input: $\mathcal{D}_{train}^{known}$ set of known members, $\mathcal{D}_{nm}^{known}$ set of non-members, integer n the size of the output datasets, $Dist$ distance between two vectors
Output: $\mathcal{D}_{train}^{Nob} \subset \mathcal{D}_{train}^{known}$ a set of members, $\mathcal{D}_{nm}^{Nob} \subset \mathcal{D}_{nm}^{known}$ non-members, minimizing N-gram bias
Require: $n \leq 1/2 * |D_{train}^{known}|$
Ensure: $n = |\mathcal{D}_{train}^{Nob}| = |\mathcal{D}_{nm}^{Nob}|$
$\mathcal{D}_{train}^{Nob} \leftarrow$ random sample of $\mathcal{D}_{train}^{known}$ of size n
$\mathcal{D}_{train}^{remain} \leftarrow \mathcal{D}_{train}^{known} \backslash \mathcal{D}_{train}^{Nob}$
$hist_{train} \leftarrow histogram(\mathcal{D}_{train}^{Nob}, \mathcal{D}_{train}^{remain})$ ▷ Compute n-gram overlap histogram
$\mathcal{D}_{nm}^{Nob} \leftarrow \emptyset$
for $i = 1$ to n **do** ▷ Select document minimizing overlap histograms distance
    $\mathcal{D}_{nm}^{remain} \leftarrow \mathcal{D}_{nm}^{known} \backslash \mathcal{D}_{nm}^{Nob}$
    $D \leftarrow \underset{D \in \mathcal{D}_{nm}^{remain}}{\arg\min} \; (d_{Kolmogorov-Sminrov}(histogram(\{D\} \cup \mathcal{D}_{nm}^{Nob}), hist_{train})$
    $\mathcal{D}_{nm}^{Nob} \leftarrow \{D\} \cup \mathcal{D}_{nm}^{Nob}$
end for

of generated verbatim text as it appears in the training data or prompts [27]. Other distance could be used, which is considered future work.

The *histo* function produces histograms of n-gram overlaps. For a given document D, which is considered as a sequence of k tokens (such as letters or words), an n-gram is defined as a continuous sequence of n tokens. The overlap of n-grams from a document D with reference to a set of documents $\mathcal{D}_{ref}$ is computed as the percentage of n-grams in D that appear in any document of $\mathcal{D}_{ref}$. The resulting histogram is a vector where each entry $hist[i]$ corresponds to the number of documents in the first dataset that have an n-gram overlap score, relative to the second dataset, equal to (or close to) value i.[9]

4.3 Constructing a Non-Classifiable Dataset

To mitigate bias indicated by the ability of agnostic classifiers to distinguish between members and non-members, we introduce the $No - Class$ algorithm (see Algorithm 2). This algorithm is designed to produce datasets where the performance of classifiers is minimized, effectively neutralizing their ability to differentiate between members and non-members.

Algorithm overview. In Algorithm 2, we consider a vector of N classifiers $(C_i)_{i \in [1,N]}$, each of which, once trained, assigns a probability in the range $[0, 1]$ to indicate the likelihood of a document being a member. The closer to 1 (respectively, 0), the more confident the classifier is that the document is a member (resp., non-member). The intuition behind our algorithm is to exploit the confidence to ensures that the constructed datasets are as challenging as possible

[9] More formally, value $hist[i]$ of the histogram is such that $hist[i] = z$ if and only if there are z documents in the first dataset with an n-gram overlap score with value between i and $i + 1$.

Algorithm 2 $No-Class$ Dataset Generation

Input $\mathcal{D}^{known}_{train}$, $\mathcal{D}^{known}_{nm}$, integer n expected cardinality of the dataset, $(C_i)_{i\in[1,N]}$ a vector of agnostic classifiers outputing $\mathbb{P}[C_i(D)]$ the probability of D being a member
Output $\mathcal{D}^{Nob}_{train} \subset \mathcal{D}^{known}_{train}$, $\mathcal{D}^{Nob}_{nm} \subset \mathcal{D}^{known}_{nm}$
Require: $n \leq 1/4 \times |D^{known}_{train}|$ & $n \leq 1/4 \times |D^{known}_{nm}|$
1: $\mathcal{D}_m \leftarrow$ random sample of $\mathcal{D}^{known}_{train}$ of size n
2: $\mathcal{D}_{nm} \leftarrow$ random sample of $\mathcal{D}^{known}_{nm}$ of size n
3: train each $(C_i)_{i\in[1,N]}$ on $\mathcal{D}_m \cup \mathcal{D}_{nm}$
4: $\mathcal{D}^{Nob}_{train} \leftarrow \emptyset$
5: $\mathcal{D}^{remain}_{train} \leftarrow \mathcal{D}^{known}_{train} \backslash \mathcal{D}_m$
6: **while** $|\mathcal{D}^{Nob}_{train}| < n$ **do** ▷ populating members minimizing confidence
7: $y \leftarrow \underset{D\in\mathcal{D}^{train}_{remain}}{\arg\min} \left\|(C_i(D) - 0.5)_{i\in[1,N]}\right\|_2$
8: $\mathcal{D}^{Nob}_{train} \leftarrow \mathcal{D}^{Nob}_{train} \cup \{y\}$
9: $\mathcal{D}^{remain}_{train} \leftarrow \mathcal{D}^{remain}_{train} \backslash \{y\}$
10: **end while**
11: $\mathcal{D}^{Nob}_{nm} \leftarrow \emptyset$
12: $\mathcal{D}^{remain}_{nm} \leftarrow \mathcal{D}^{known}_{nm} \backslash \mathcal{D}_{nm}$
13: **while** $|\mathcal{D}^{Nob}_{nm}| < n$ **do** ▷ populating non-members minimizing confidence
14: $y \leftarrow \underset{D\in\mathcal{D}^{nm}_{remain}}{\arg\min} \left\|(C_i(D) - 0.5)_{i\in[1,N]}\right\|_2$
15: $\mathcal{D}^{Nob}_{nm} \leftarrow \mathcal{D}^{Nob}_{nm} \cup \{y\}$
16: $\mathcal{D}^{remain}_{nm} \leftarrow \mathcal{D}^{remain}_{nm} \backslash \{y\}$
17: **end while**

for the classifiers. It is worth noting that other variants of this algorithm have been implemented, balancing the number of false positives, false negatives, true positives, and true negatives in each member/non-member class. The algorithm operates in the two main steps:

- *Sampling and training:* The algorithm begins by randomly sampling known members and non-members from the dataset (lines 1-2). These samples are then used to train a set of N agnostic classifiers $(C_i)_{i\in[1,N]}$ (line 3).
- *Confidence minimization:* Using the classifiers $(C_i)_{i\in[1,N]}$ on the left-out members, we then construct $\mathcal{D}^{Nob}_{train}$ (lines 6 to 10) and $\mathcal{D}^{Nob}_{nm}$ (lines 13 to 17) minimizing the overall confidence of the classifiers. Since the further $C_i(D)$ is from 0.5, the more confident C_i is in its assessment of document D, we consider the vector $(C_i(D) - 0.5)_{i\in[1,N]}$ representing the confidence of each classifiers. At each step, we add the element D that minimizes the l2-norm of this confidence vector. For instance, considering the construction of considered members: (1) among the members that have neither been selected as "Non-biased" (Nob) nor used to train the classifiers ($D \in \mathcal{D}^{remain}_{train}$), the one that minimizes the l2-norm of the confidence vector (i.e., $\left\|(C_i(x) - 0.5)_{i\in[1,N]}\right\|_2$) is selected (line 7); (2) the aforementioned element is inserted in the set (line 8); (3) the set of remaining candidates is updated (line 9).

5 Experimental validation of our approach

In this section, we apply and evaluate our proposal with reference to the Gutenberg dataset. Experimental settings are described in Sec. 5.1, detailing how the candidate datasets are constructed to avoid a priori bias, how bias are assessed, as well as the MIAs and LLMs assessed on the datasets produced following our proposal. In spite of known members and non-members being constructed to circumvent bias, Sec. 5.2 shows that random samples exhibit n-gram bias. Such bias are addressed in Sec 5.3 by producing datasets following the $No-Ngram$ algorithm (Alg. 1), which still exhibit residual bias exploitable by an agnostic classifier. Section 5.4 assesses the last pair of datasets produced following the $No-Class$ algorithm (Alg. 2). Finally, Sec. 5.5 presents the assessment of MIAs using the produced datasets and discusses the impact of bias in the evaluation of MIAs. All the code, resulting analysis and dataset are available online[10].

5.1 Experimental Setting

Dataset: Gutenberg Project. The project Gutenberg[11] offers a high-quality open dataset of over 70,000 books, continuously expanding. PG-19 [23], a subset of 28,752 books extracted in 2019, has been included in RedPajama-Data [5] and The Pile [10] and used to train LLMs such as Pythia [1] and OpenLLaMA [11]). It is also widely used to evaluate MIAs (e.g. [19], [18]). We use documents from Project Gutenberg in our experiments because of its quality, recognized relevance in MIA research and the availability of methods [19] to minimize bias.

We assume an LLM trained on PG-19 [23] and draw our *members* from this dataset. Regarding *non-members*, since Project Gutenberg is ongoing, with books continuously added, all english books added after the publication of PG-19 are potential non-members. To circumvent the potential for temporal bias between the member and non-member sample, we adhere to the methodology proposed by Meeus et al. [19] and restrict our analysis to books published between 1850 and 1910. This leads to final sets $\mathcal{D}_{\text{train}}^{known}$ and $\mathcal{D}_{nm}^{known}$ of 7300 and 2400 books, respectively. Note also that Das et al. [6] identified a potential bias in datasets constructed following this methodology. Indeed, they showed that the format of the preface metadata that project Gutenberg adds to books has changed since 2019. To circumvent this, we discard such metadata. Therefore, our starting sets of members $\mathcal{D}_{\text{train}}^{known}$ and non-members $\mathcal{D}_{nm}^{known}$ are chosen because, **a priori, there is no (known) bias affecting them.**

Bias assesment. The *agnostic classifiers* employed in this study utilize a Bayes algorithm for multinomially distributed data, focusing on the distribution of 1 to 3-grams. These classifiers are trained and applied using scikit-learn, chosen for its robustness in handling such data distributions. For the *n-gram analysis*, characters are treated as tokens when computing n-grams. We focus on n=7, as

[10] https://github.com/ceichler/MIA-bias-removal
[11] https://www.gutenberg.org/

previous research [7] has shown that 7-grams reveal the most significant distributional differences. The n-gram analysis is conducted using a Bloom filter based on the implementation of [12], ensuring efficient and accurate bias detection.

LLMs. We conduct our experiments on two autoregressive large language models: OpenLLaMA [11] and Pythia [1]. OpenLLaMA is a series of 3B, 7B and 13B open-source models trained on 1T tokens that aims to emulate Meta's LLaMA [28]. OpenLLaMA is trained on RedPajama-Data [5], an open-source reproduction of the original LLaMA training dataset. Pythia is an open and transparent suite of LLMs ranging in size from 70M to 12B parameters that has been specifically released to enable research. The language models in Pythia have been trained on The Pile [10]. In this work, we have used the `OpenLLaMA-3B` and `Pythia-2.8B` models. Both The Pile and RedPajama-Data include PG-19 [23].

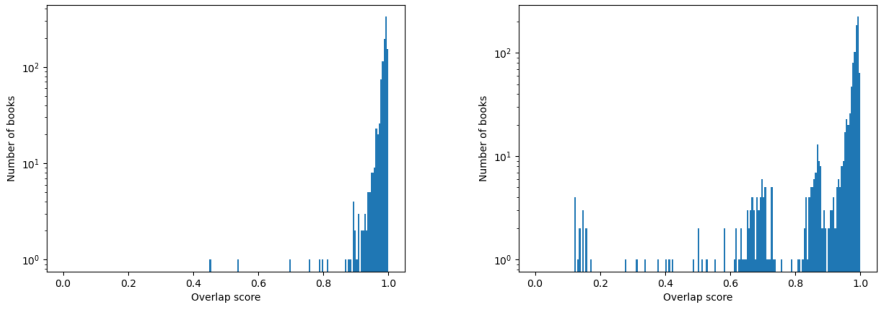
MIAs. We conduct our experiments adapting the codes provided by [18] with the following state-of-the-art MIAs:

- **Min-k% Prob.** This metric is based on the likelihood of the $k\%$ of tokens in a sequence D that have the lowest probabilities, based on the preceding tokens [25].
- **Max-k% Prob.** This is the inverse metric of Min-k% Prob, based on the tokens that have the highest probabilities. We use $k = 10$ for both Min-k% Prob and Max-k% Prob.
- **zlib Ratio.** This MIA identifies potential member when having a low ratio of the model's perplexity to the entropy of the text [2]. This entropy is calculated as the number of bits required to compress the sequence using the zlib library [8].
- **Perplexity (ppl).** This MIA leverages perplexity [2] as scores and then threshold them to classify samples as members or non-members.
- **Meta_MIA.** This MIA is based on the work of [18], which aggregates 52 MIAs (including Min-k% Prob, perplexity, zlib Ratio, etc.) to create a single feature vector. A linear regressor is trained to learn the importance of weights for the different MIA attacks and thus classify their membership status.

5.2 Assuming No bias: Random Sample

By construction, $\mathcal{D}_{train}^{known}$ and $\mathcal{D}_{nm}^{known}$ are exempt of bias related to meta-data and time-shift. Since there is no reason to suspect a bias, we construct $\mathcal{D}_{train}^{NoB}$ and $\mathcal{D}_{nm}^{NoB}$ through a random sample. We compute the distribution of n-gram overlap of these two sets with reference to the left out members ($\mathcal{D}_{train}^{known} \backslash \mathcal{D}_{train}^{NoB}$) as described in Sec. 4.2. The resulting histograms are depicted in Fig. 1.

Surprisingly, in-spite of the absence of time-shift and metadata bias, the set of members and non-members exhibit significant distributional difference of n-gram overlap, with a KS distance of 0.222. This bias can be exploited by an agnostic classifier achieving an AUC ROC of 0.84 (reported hereafter).



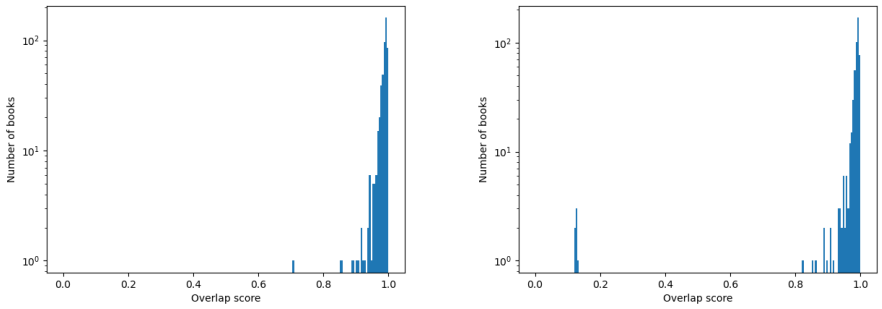
(a) Randomly selected members (b) Randomly selected non-members

Fig. 1: Histograms of n-gram overlap wrt left out members (KS-distance = 0.222)

Conclusion. The text of books written in the same time interval but added to project Gutenberg at different dates (before or after the extraction of PG-19) still exhibit n-gram shifts and a random sampling produce heavily biased datasets.

5.3 $No-Ngram$ to Minimize N-gram Bias

To address the highlighted n-gram overlap bias, we produce new samples $\mathcal{D}_{train}^{Nob}$ and $\mathcal{D}_{nm}^{Nob}$ following the $No-Ngram$ algorithm. The corresponding distributions of n-gram overlap are depicted in Fig. 2. Their KS distance is 0.034, a drastic 84% drop from 0.222, the distance achieved with random samples. Notably, non-members exhibit high n-gram overlap with the left out members, only 5 non-member books having an overlap score lesser than 0.8.

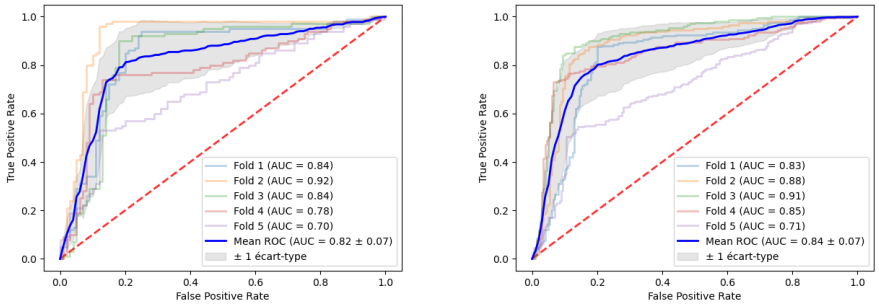


(a) $No-Ngram$ members (b) $No-Ngram$ non-members

Fig. 2: Histograms of n-gram overlap wrt left out members (KS-distance = 0.034)

We further assess residual bias by training a classifier on $\mathcal{D}_{train}^{NoB}$ and $\mathcal{D}_{nm}^{NoB}$. The ROC of each fold is illustrated in Fig. 3a. As a reference, the evaluation

of a classifier trained on randomly sampled datasets is depicted in Fig. 3b. The agnostic classifier achieves on average over 5 folds 0.82 AUC ROC and 5%, 26%, and 49%TPR at 1%, 5%, and 10% FPR, respectively. This remains highly accurate and the accuracy loss is marginal when compared to random samples where an agnostic classifier achieves 0.84 AUC ROC and 3%, 30%, and 63% TPR at 1%, 5%, and 10% FPR, respectively.



(a) trained on $No - Ngram$ sets (b) trained on randomly sampled set

Fig. 3: ROC of 5-folds agnostic classifiers

Conclusion. While the $No - Ngram$ algorithm has successfully reduced (if not eliminated) the bias in n-gram overlap, the produced sets can still be discriminated with high accuracy by an agnostic classifier. Contrarily to previous proposal [7], this suggests that distributional difference in n-gram overlap is insufficient as a metric of MIA benchmark difficulty.

5.4 Sampling unclassifiable datasets

The datasets being classifiable even when n-gram bias are minimized, we apply $No - Class$[12] using a single classifier trained on randomly sampled set whose evaluation is presented in Fig. 3b. To evaluate residual bias, we train a new agnostic classifier on the resulting sets whose evaluation is shown in Fig. 4.

On average, the agnostic classifier achieves 6%, 13%, and 20%TPR at 1%, 5%, and 10% FPR, respectively. Interestingly, the 5th fold achieves lower TPR than a random guess for FPR in the 30-45% interval. Overall, performance is slightly better than random, particularly at low FPR, but significantly worse than previous settings. Indeed, the TPR at 5% and 10% FPR decrease by roughly 56% and 72% when compared to a classifier trained on random samples, respectively. Similarly, the AUC ROC drops from 0.84 to 0.58, denoting a 76% decrease of the distance to the AUC ROC of a random guess.

Conclusion. These results indicate that hard-to-classify sets also resist training, showing minimal exploitable bias for agnostic classifiers.

[12] Since we use a single classifier, we also ensure the same number of false positive, false negative, true positive and true negative in the selected sets.

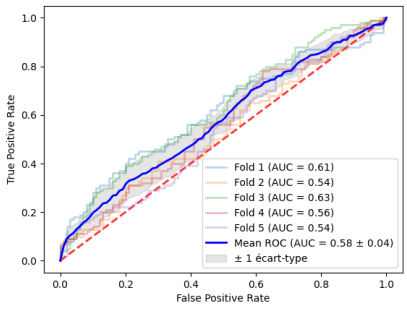


Fig. 4: ROC of 5-folds agnostic classifier trained on $No-Class$ sets

Model	Dataset	Meta_MIA	Max10%Prob	ppl	zlib_ratio	Max10%Prob
`Pythia-2.8B`	random	0.314	0.085	0.087	0.111	0.062
`Pythia-2.8B`	$No-Ngram$	0.350	0.149	0.050	0.145	0.124
`Pythia-2.8B`	$No-Class$	**0.314**	**0.313**	0.127	0.140	0.162
`OpenLLaMA-3B`	random	0.371	0.111	0.059	0.139	0.122
`OpenLLaMA-3B`	$No-Ngram$	0.386	0.156	0.074	0.131	0.222
`OpenLLaMA-3B`	$No-Class$	**0.224**	0.070	0.046	0.132	0.081

Table 1: TPR values at 10%FPR. Bold values outperform agnostic classifiers.

5.5 Assesment of MIAs

We assess 6 state of the art MIAs on 2 LLMs and the datasets random, $No-Ngram$, and $No-Class$ presented Sec. 5.2, 5.3, and 5.4, respectively. Tables 1 and 2 report their TPR at 10%FPR and AUC ROC averaged over 5 runs.

Overall, no MIA manage to outperform an agnostic classifier on the random and $No-Ngram$ datasets. Only Meta-MIA outperforms the classifier on $No-Class$ on both `OpenLLaMA-3B` and `Pythia-2.8B`, while 10%_min_probs outperforms it solely on `Pythia-2.8B` according to the TPR@10%FPR.

Meta-MIA is consistently significantly above a random guess and the best MIA across all settings and metric. It achieves its best results on `Pythia-2.8B`, with 37.1%TPR@10%FPR and a AUC ROC of 0.74 for the random biased dataset. On $No-Class$, these values drop to 22.4% and 0.634, denoting a ratio $No-Class$/random of 0.60 and 0.857.

Conclusion. Meta-MIA is consistently the best out of the 6 MIAs evaluated on our datasets. Yet, its TPR@10%FPR and AUC ROC drop by 40% and 14.3% respectively when evaluated on datasets produced using our approach rather

Model	Dataset	Meta_MIA	Min10%Prob	ppl	zlib_ratio	Max10%Prob
`Pythia-2.8B`	random	0.692	0.544	0.554	0.494	0.490
`Pythia-2.8B`	$No-Ngram$	0.688	0.538	0.477	0.544	0.557
`Pythia-2.8B`	$No-Class$	0.670	0.665	0.583	0.531	0.578
`OpenLLaMA-3B`	random	0.740	0.523	0.493	0.501	0.576
`OpenLLaMA-3B`	$No-Ngram$	0.744	0.543	0.545	0.506	0.642
`OpenLLaMA-3B`	$No-Class$	**0.634**	0.520	0.503	0.523	0.494

Table 2: AUC ROC values. Bold values outperform an agnostic classifier.

than on randomly sampled ones. This underlines the importance of our approach to accurately estimate MIAs performances.

6 Conclusion

As LLMs are trained leveraging myriads of data items, including copyrighted ones, it is key to ascertain whether a piece of data has been used in this process. Yet, the effectiveness of MIAs has been recently questioned, due to the existence of biases in datasets constructed ex-post. This work introduces $Nob-MIAs$, a set of algorithms to build unbiased datasets, thus setting a more solid ground for MIA assessment. Our experiments on the Gutenberg dataset confirms that our approach significantly reduces bias (e.g., an 84% reduction of difference in n-gram overlap distribution) and impacts on MIA evaluation, with TPR@10%FPR and ROC AUC of the best-performing MIA (out of 6) decreasing by 40% and 14.3% respectively, compared to evaluations on randomly sampled datasets.

This work opens several future research avenues, including extending the algorithms to detect and mitigate residual biases and applying this approach to non-textual MIAs, where ex-post dataset construction is also common.

Acknowledgments. This work was supported by the French grant iPoP PEPR (ANR-22-PECY-0002) and DATAIA. Jose Maria de Fuentes has also received support from the Spanish National Cybersecurity Institute (INCIBE) grant APAMciber within the framework of the Recovery, Transformation and Resilience Plan funds, financed by the European Union (Next Generation); and from UC3M's Requalification programme, funded by the Spanish Ministerio de Ciencia, Innovacion y Universidades with EU recovery funds (Convocatoria de la Universidad Carlos III de Madrid de Ayudas para la recualificación del sistema universitario español para 2021-2023, de 1 de julio de 2021).

References

1. Biderman, S., Schoelkopf, H., Anthony, Q., Bradley, H., O'Brien, K., Hallahan, E., Khan, M.A., Purohit, S., Prashanth, U.S., Raff, E., Skowron, A., Sutawika, L., Van Der Wal, O.: Pythia: a suite for analyzing large language models across training and scaling. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023)
2. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2633–2650 (2021)
3. Chang, K.K., Cramer, M., Soni, S., Bamman, D.: Speak, memory: An archaeology of books known to chatgpt/gpt-4. arXiv preprint arXiv:2305.00118 (2023)
4. Cheng, J., Marone, M., Weller, O., Lawrie, D., Khashabi, D., Van Durme, B.: Dated data: Tracing knowledge cutoffs in large language models. arXiv preprint arXiv:2403.12958 (2024)
5. Computer, T.: Redpajama-data: An open source recipe to reproduce llama training dataset (2023), `https://github.com/togethercomputer/RedPajama-Data`

6. Das, D., Zhang, J., Tramèr, F.: Blind baselines beat membership inference attacks for foundation models. arXiv preprint arXiv:2406.16201 (2024)
7. Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., Hajishirzi, H.: Do membership inference attacks work on large language models? CoRR **abs/2402.07841** (2024). https://doi.org/10.48550/ARXIV.2402.07841, https://doi.org/10.48550/arXiv.2402.07841
8. Gailly, J.l., Adler, M.: Zlib compression library (2004)
9. Galli, F., Melis, L., Cucinotta, T.: Noisy neighbors: Efficient membership inference attacks against llms. arXiv preprint arXiv:2406.16565 (2024)
10. Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., Leahy, C.: The pile: An 800gb dataset of diverse text for language modeling (2020), https://arxiv.org/abs/2101.00027
11. Geng, X., Liu, H.: Openllama: An open reproduction of llama (May 2023), https://github.com/openlm-research/open_llama
12. Groeneveld, D., Ha, C., Magnusson, I.: Bff: The big friendly filter (2023), https://github.com/allenai/bff
13. Jedrzejewski, F.V., Thode, L., Fischbach, J., Gorschek, T., Mendez, D., Lavesson, N.: Adversarial machine learning in industry: A systematic literature review. Computers & Security p. 103988 (2024). https://doi.org/https://doi.org/10.1016/j.cose.2024.103988, https://www.sciencedirect.com/science/article/pii/S0167404824002931
14. Kaneko, M., Ma, Y., Wata, Y., Okazaki, N.: Sampling-based pseudo-likelihood for membership inference attacks. arXiv preprint arXiv:2404.11262 (2024)
15. Li, H., Deng, G., Liu, Y., Wang, K., Li, Y., Zhang, T., Liu, Y., Xu, G., Xu, G., Wang, H.: Digger: Detecting copyright content mis-usage in large language model training. arXiv preprint arXiv:2401.00676 (2024)
16. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
17. Liu, X., Sun, T., Xu, T., Wu, F., Wang, C., Wang, X., Gao, J.: Shield: Evaluation and defense strategies for copyright compliance in llm text generation. arXiv preprint arXiv:2406.12975 (2024)
18. Maini, P., Jia, H., Papernot, N., Dziedzic, A.: Llm dataset inference: Did you train on my dataset? arXiv preprint arXiv:2406.06443 (2024)
19. Meeus, M., Jain, S., Rei, M., de Montjoye, Y.: Did the neurons read your book? document-level membership inference for large language models. In: Balzarotti, D., Xu, W. (eds.) 33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024. USENIX Association (2024), https://www.usenix.org/conference/usenixsecurity24/presentation/meeus
20. Meeus, M., Jain, S., Rei, M., de Montjoye, Y.A.: Inherent challenges of post-hoc membership inference for large language models. arXiv preprint arXiv:2406.17975 (2024)
21. Meeus, M., Shilov, I., Faysse, M., de Montjoye, Y.A.: Copyright traps for large language models. In: Forty-first International Conference on Machine Learning (2024), https://openreview.net/forum?id=LDq1JPdc55
22. Panaitescu-Liess, M.A., Che, Z., An, B., Xu, Y., Pathmanathan, P., Chakraborty, S., Zhu, S., Goldstein, T., Huang, F.: Can watermarking large language models prevent copyrighted text generation and hide training data? arXiv preprint arXiv:2407.17417 (2024)
23. Rae, J.W., Potapenko, A., Jayakumar, S.M., Lillicrap, T.P.: Compressive transformers for long-range sequence modelling (2019), https://arxiv.org/abs/1911.05507

24. Reuel, A., Bucknall, B., Casper, S., Fist, T., Soder, L., Aarne, O., Hammond, L., Ibrahim, L., Chan, A., Wills, P., et al.: Open problems in technical ai governance. arXiv preprint arXiv:2407.14981 (2024)
25. Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., Zettlemoyer, L.: Detecting pretraining data from large language models. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=zWqr3MQuNs
26. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)
27. Sonkar, S., Baraniuk, R.G.: Many-shot regurgitation (msr) prompting. arXiv preprint arXiv:2405.08134 (2024)
28. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
29. Wei, J.T.Z., Wang, R.Y., Jia, R.: Proving membership in llm pretraining data via data watermarks. arXiv preprint arXiv:2402.10892 (2024)
30. Yan, B., Li, K., Xu, M., Dong, Y., Zhang, Y., Ren, Z., Cheng, X.: On protecting the data privacy of large language models (llms): A survey. arXiv preprint arXiv:2403.05156 (2024)
31. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. In: 2018 IEEE 31st computer security foundations symposium (CSF). pp. 268–282. IEEE (2018)