Scientific Research Publishing

Journal of Information Security, 2024, 15, 448-473
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

Protecting LLMs against Privacy Attacks While Preserving Utility

Gunika Dhingra[1], Saumil Sood[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, Georgia, USA
Email: 12gunika@gmail.com, sood.saumil03@gmail.com, zeba.wase@gmail.com, arshdeep@cloudemy.io, madisetti.vijay@gmail.com

How to cite this paper: Dhingra, G., Sood, S., Wase, Z.M., Bahga, A. and Madisetti, V.K. (2024) Protecting LLMs against Privacy Attacks While Preserving Utility. *Journal of Information Security*, **15**, 448-473.
https://doi.org/10.4236/jis.2024.154026

Received: August 3, 2024
Accepted: September 17, 2024
Published: September 20, 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. This inadvertent leakage of sensitive information typically occurs when the models are subjected to black-box attacks. To address the growing concerns of safeguarding private and sensitive information while simultaneously preserving its utility, we analyze the performance of Targeted Catastrophic Forgetting (TCF). TCF involves preserving targeted pieces of sensitive information within datasets through an iterative pipeline which significantly reduces the likelihood of such information being leaked or reproduced by the model during black-box attacks, such as the autocompletion attack in our case. The experiments conducted using TCF evidently demonstrate its capability to reduce the extraction of PII while still preserving the context and utility of the target application.

Keywords

Large Language Models, PII Leakage, Privacy, Memorization, Membership Inference Attack (MIA), Defenses, Generative Adversarial Networks (GANs), Synthetic Data

1. Introduction

Large Language Models are currently at the forefront of discussions in the field of artificial intelligence and its subsequent systems. These advanced neural networks possess the capability to comprehend and generate human level language by

analyzing extensive data [1]. To achieve their current proficiency, these networks are trained on colossal datasets equipping them with the proficiency to not only understand and produce natural language but also to handle diverse content across various tasks [2].

During the training process, a wide range of data sources, including web-scraped data and confidential repositories, are used. These sources often contain private, sensitive information, such as Personally Identifiable Information (PII), including names, addresses, locations, and email addresses which potentially serve as direct identifiers of individuals or institutions. While training, LLMs inadvertently assimilate this sensitive data, a process facilitated by the mechanism of memorization [3], which has the potential to occur through the training as well as the fine-tuning process. Consequently, when these models are specifically prompted—often with the intent to extract sensitive information—there exists a substantial risk that they might accidentally disclose such details.

There appear to have been numerous documented instances where LLMs, either directly or indirectly, utilize the intellectual property data of various institutions. A notable lawsuit [4] filed by the New York Times against OpenAI and Microsoft highlights this issue which alleges that OpenAI had used millions of copyrighted NYT articles to train its generative AI models, including ChatGPT. It also reflected an even increased leakage of verbatim information by the Microsoft Copilot. The New York Times contends that OpenAI's AI models are capable of "memorizing" parts of the copyrighted works included in their training datasets, occasionally generating near-verbatim reproductions. Such instances underscore the problematic nature of memorization, especially when involving web-scraped content that infringes on intellectual property rights.

While LLMs learn from various sources, they become repositories of vast amounts of information, including sensitive or confidential data, posing inherent risks of unauthorized extraction. Vulnerabilities such as backdoor attacks [5], membership inference attacks, and model inversion attacks enable attackers to extract sensitive data from both pre-trained and fine-tuned models. Our study focuses on a prevalent form of such attacks: the autocomplete attack. This technique exploits the model's predictive power by providing minimal, partial prompts that encourage the model to complete or add to the given information. Consequently, this process can inadvertently reproduce memorized data including sensitive, private, or confidential information like PII.

This potential for sensitive information leakage highlights the immediate need for robust privacy and safeguarding mechanisms. One promising solution is unlearning which involves techniques to remove or reduce specific knowledge from the model without retraining it from scratch. Popular methods include fine-tuning based, gradient-based unlearning [6], selective pruning unlearning [7], and prompting for unlearning [8]. However, these methods often have significant drawbacks, such as being computationally expensive, requiring extensive model access, instability, and lack of guarantees.

To address these challenges, we propose a novel technique called Targeted Catastrophic Forgetting (TCF). This technique iteratively targets and modifies PII in a dataset to protect the original sensitive information. TCF employs multiple rounds of successive refinements and utilizes three distinct sub-approaches to protect essential private information. For each round and approach, contextually relevant synthetic datasets are generated using two distinct resources: a Conditional Generative Adversarial Network (CGAN) [9] and an LLM (specifically, GPT-4 accessed via the OpenAI API).

We use CGANs for text generation due to their ability to produce contextually relevant and coherent text by conditioning the generation process on specific inputs. This conditioning allows CGANs to generate more precise and diverse outputs compared to traditional GANs, which lack such specificity. Unlike other methods such as Variational Autoencoders (VAEs) or standard autoregressive models, CGANs leverage an adversarial training mechanism, enhancing the quality and realism of the generated text through a dynamic feedback loop between the generator and discriminator. Additionally, we particularly utilize OpenAI's GPT-4 model for its proven high capability in synthetic data generation matching human like capabilities.

The following sections provide a thorough examination of this proposed technique and the experiments involved, emphasizing the effectiveness of TCF in reducing the risk of PII leakage while maintaining the utility of the generated data.

2. Related Works

This section explores several resonant studies that address similar situations, forming the foundation of this paper. By examining these relevant works, we gain insights into the current landscape of data privacy challenges and solutions related to Large Language Models.

With the rapid expansion of large language models (LLMs), data privacy concerns are also intensifying. A noteworthy aspect is that LLMs rely on extensive amounts of training data, including texts scraped from the internet, publicly available datasets, and proprietary sources, which raises significant privacy concerns, particularly concerning Personally Identifiable Information (PII). An exemplary study [10], which is a comprehensive survey of data privacy concerns related to LLMs, delineates the spectrum of data privacy threats, encompassing both passive privacy leakage and active privacy attacks within LLMs. Another recent study [5] further elucidates these concerns by quantifying the risks associated with backdoor membership inference attacks and PII-focused attacks on LLMs, thereby shedding more light on the critical issue of information assimilation and privacy preservation.

Additionally, another remarkable study [11] explores the ethical challenges arising from security threats to LLMs, scrutinizing five major threats or vulnerabilities, ranging from jailbreaking LLMs to prompt injections, with a particular focus on PII leaks. Likewise to other studies, it also underscores that LLMs, trained

on vast amounts of web data, encompass sensitive PII, which the models learn through memorization techniques and can potentially be uncovered by attackers. Vijay Madisetti and Arshdeep Bahga proposed a novel technique, called Targeted Catastrophic Forgetting (TCF) in 2023, that aligns with this spectrum, specifically targeting PII to safeguard it from potential leakage by the models. This framework makes an attempt at fostering trust in these language models while maintaining the utility they are designed for.

We have also explored how GANs have transformed generative modeling, yet frequently face challenges in producing text that aligns with specific categories. [12]. Recent research introduces Category-aware Generative Adversarial Networks (CatGAN) [13], a framework that uses hierarchical evolutionary learning to improve category-specific text generation. This method incorporates category information directly into the adversarial process and uses evolutionary algorithms to refine both the generator and discriminator, resulting in high-quality, contextually appropriate text. This aided in understanding challenges such as training instability and offers insights for implementing and varying conditions in our use case.

Moreover, studies shows that catastrophic forgetting is a common issue that worsens with increasing model size [14]. Architectural differences, such as those between decoder-only and encoder-decoder models, affect the degree of forgetting. The analysis also notes that LLMs can reduce certain biases during continual fine-tuning. These findings highlight the need for effective strategies to prevent forgetting, enabling LLMs to expand their knowledge without losing previously learned information. This understanding is crucial for developing and improving our approach of TCF.

3. Overview of Our Paper

The principal aim of this study is to analyze the performance of a method for protecting privacy of LLMs called Targeted Catastrophic Forgetting (TCF), first proposed by Madisetti and Bahga, in 2023, as part of several US Patent applications. Additionally, we conduct a comprehensive evaluation of the performance of an LLM that has been fine-tuned on a richly detailed dataset, with a focus on its vulnerability to a privacy leakage attack that poses a risk for exposing the sensitive and confidential information, including personally identifiable information (PII). To thoroughly assess the efficiency of our proposed technique, we conduct the experimentation via three different procedures which have been discussed elaborately in the following sections.

3.1. Targeted Catastrophic Forgetting

Targeted catastrophic forgetting in large language models is the intentional and iterative process of making a model forget specific knowledge (in our case the PII or any other sensitive information) by retraining it with new yet contextually similar data that conflicts with the targeted information, aiming to erase it from the

memory of the language model while keeping the model's capabilities and performance intact.

3.1.1. Theoretical Basis and Working Principle of TCF

Figure 1 illustrates the iterative fine-tuning process used in TCF with autocompletion attack evaluation. The process begins with a base model that undergoes fine-tuning and adapter merging, followed by response generation under autocompletion attack. The generated responses are evaluated using various metrics, and comparisons are made across different iterations to assess performance. TCF builds upon the concept of catastrophic forgetting, a phenomenon where neural networks tend to abruptly forget previously learned information when learning new data. We leverage this principle strategically to "overwrite" sensitive information while preserving general knowledge and capabilities. The working principle of TCF involves multiple rounds of iterative fine-tuning, utilizing a progressive approach to dataset generation that escalates from a base to a soft, and eventually to a hard privacy approach. This process equips models with the ability to target and prevent the generation of sensitive data points (PII) when compared to the actual available information used in training/fine-tuning. By preventing the generation of such data, TCF establishes a protective mechanism against potential leakage during black-box attacks, such as the autocomplete attack examined in our study. The iterative and targeted fine-tuning process enhances the model's coherence, ensuring that utility is maintained while privacy is effectively induced.

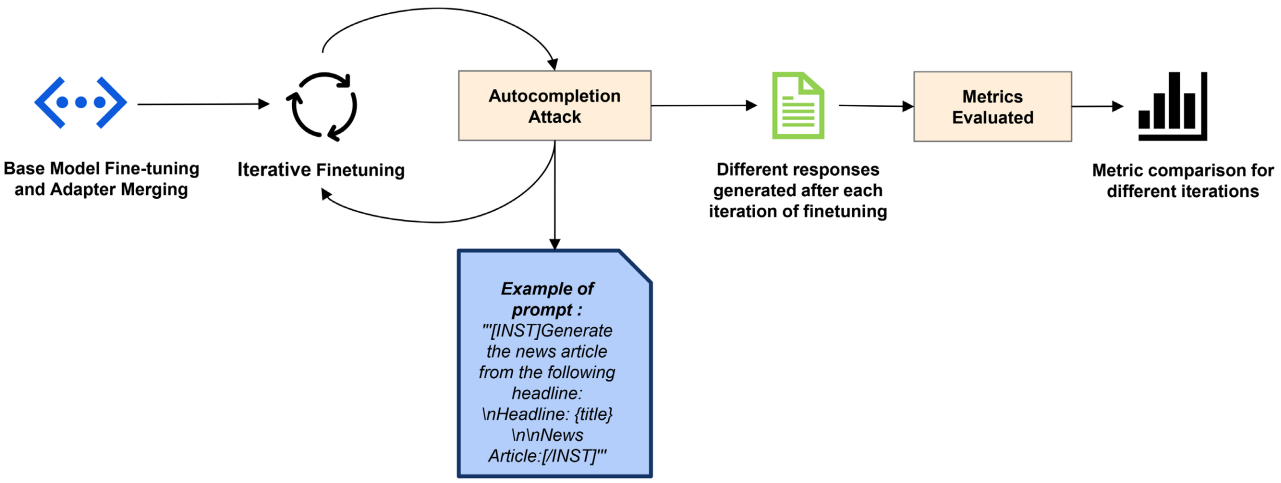

Figure 1. Overview of the iterative fine-tuning process in TCF with auto-completion attack evaluation.

The TCF approach involves identifying the specific information that needs to be forgotten, creating datasets by prompting large language models (LLMs) or generating synthetic data using Generative Adversarial Networks (GANs) to replace the personally identifiable information (PII). This process is conducted across three different sub-approaches with a ranging degree of privacy and the extent of safeguarding sensitive information. We then fine-tune the model with these newly created datasets and evaluate the results to ensure that the sensitive

or confidential information is no longer present while still preserving the overall utility and performance of the model as shown in Figure 2. The idea of this approach is that when we iteratively target the PII and make the model learn different data points for the same context, then eventually the model potentially shall not leak these essential pieces of sensitive and private information when subjected to even black-box attacks, in our case the autocompletion attack.

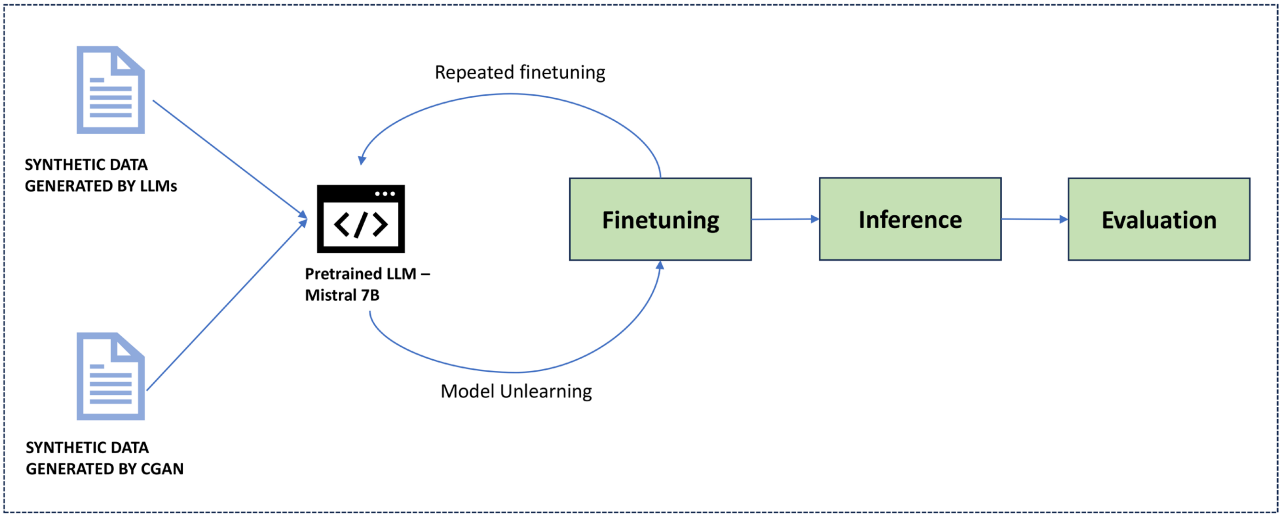

Figure 2. This figure illustrates the process of targeted catastrophic forgetting in large language models (LLMs) to remove specific knowledge, such as personally identifiable information (PII), while preserving the model's overall capabilities and performance.

3.1.2. Protecting Sensitive Information in the Training Process

TCF specifically targets the PII prevalent in the text used in successive refinement rounds. These targeted data points are replaced by their counterparts using two mechanisms: Generative Adversarial Networks (GANs) and GPT-4. This replacement strategy ensures that models remain contextually relevant while making substantial progress in preserving sensitive information.

3.1.3. Quantitative Indicators of Privacy Protection

Once the models are finetuned, they undergo an auto-completion attack, where each respective model is tasked with generating a news article based on a given headline. All generated outputs are saved and subjected to these metrics to rigorously assess the performance of the models and, consequently, the effectiveness of our approach. These metrics provide a detailed evaluation, allowing us to analyze various aspects of the model's performance, including accuracy, consistency, and contextual relevance. Specifically, we use the Extraction Success Rate (ESR) which measures the proportion of unique personally identifiable information (PII) entities extracted by the language. Additionally, we use the Cosine Similarity and Jaccard Similarity measures to compare the similarity between two pieces of text. Details on the metrics are provided in section 6.

3.1.4. Black-Box Attack Simulation and Testing Methods

Our study focused on auto-completion attacks, where the model is prompted to

complete partial information, potentially revealing sensitive data. This attack vector is common in real-world scenarios where adversaries attempt to extract information through seemingly innocuous queries.

3.2. Proposed Pipeline

The proposed pipeline involves multiple rounds of finetuning using datasets generated by both GAN and LLM for testing and training purposes. This iterative process, referred to as successive refinements, aims to enhance model performance progressively. In each round, the model is refined using its respective dataset and subsequently tested through an autocompletion attack, tasked with generating news articles based on provided headlines. To ensure robust evaluation, the model generates two articles per headline prompt, evaluated using a comprehensive set of metrics detailed in Section 9. Performance scores are derived by averaging the results from each pair of generated outputs, providing an overall measure of the model's effectiveness in each refinement round.

The pipeline encompasses three distinct approaches to finetuning: Approach 1 establishes a baseline where the Mistral 7B model undergoes four rounds of refinement using independently curated datasets from both GAN and LLM. This method focuses on progressively enhancing the model's performance. Approach 2 extends upon the baseline by employing a more informed LLM for finetuning, starting with the refined model from Approach 1's first round to deepen contextual understanding and mitigate privacy risks over five rounds. Approach 3 further advances the refinement process by utilizing the highly informed model from Approach 2's fourth round, iteratively enhancing it over five rounds with GAN and LLM-generated data. This approach aims to achieve heightened contextual relevance and informativeness compared to Approach 2, emphasizing the iterative evolution and robust evaluation of model performance across multiple rounds of refinement.

3.3. Dataset Generation

For the scope of this study, we have generated multiple sets of synthetic dataset, the creation of which was informed by the need to investigate the potential for verbatim leakage of NY Times articles. This experimentation was prompted by a lawsuit filed by the NY Times against OpenAI and Microsoft, which alleges that advanced models like ChatGPT, utilizing the GPT-4 architecture, and Microsoft Copilot, have reproduced NY Times articles verbatim, thereby infringing on intellectual property rights. Consequently, we crafted a dataset mirroring these circumstances, where each data instance comprises the title of an article, along with its first and second paragraphs. The dataset was generated to replicate the real-world scenario while also checking for potential leakage of personally identifiable information (PII), as well as private, confidential, and sensitive information by the models post the proposed techniques. In our research, we utilize two distinct methods for generating data: the first involves the use of Generative Adversarial

Networks (GANs), and the second uses utilizes the GPT-4 large language model to create synthetic datasets. Extensive details on each methodology are provided below, offering insight into the mechanisms and implications of each approach in the context of our study.

3.3.1. Dataset Generation Using GAN

GANs offer significant advantages for text generation due to their adversarial training mechanism, which promotes the creation of high-quality, realistic text. By leveraging a generator and a discriminator in a competitive setup, GANs can produce diverse and coherent text that captures the nuances of human language. Additionally, CGANs which we have used in our study allow for controlled and domain-specific text generation, making them highly adaptable.

The architecture of CGAN comprises a generator and discriminator, each pivotal in the adversarial training process for enhanced output control. The generator begins with inputs including conditional attributes and Gaussian noise vectors, which add stochasticity and prevent overfitting to specific training data patterns. Careful adjustment ensures these inputs do not disrupt coherence or semantic meaning, influencing the style and variation of the generated text. These inputs are processed through layers that concatenate them effectively, followed by transformations through fully connected and convolutional layers. Batch normalization stabilizes training, while activation function ReLU ensures non-linearity, culminating in a tanh activation at the output to constrain generated outputs within a specific range. Whereas, the discriminator receives the same conditional input as the generator, aiding in the classification of real and generated data samples. Its architecture involves concatenated inputs processed through connected layers, normalized for stability, and activated with the function Leaky ReLU. The final layer outputs a probability score indicating whether the input data sample is real (close to 1) or generated by the generator (close to 0) Adversarial training harmonizes these components: the generator strives to produce outputs that deceive the discriminator into categorizing them as real, while the discriminator refines its ability to differentiate between original and synthetic data.

CGANs are the types of networks that can be constructed by merely feeding the extra auxiliary information that is text, which extends the GAN into CGAN. Generator of CGAN takes this text and a latent vector z so that it generates conditional real-looking data $G(z|c)$, and discriminator of CGAN takes the extra auxiliary information c and real data x, so that it distinguishes generator generated samples $D(G(z|c))$ from real data x. CGAN can control the generation of data, which is impossible with plain GANs. The loss functions for CGANs is as follows:

$$L_{CGAN} = \mathbb{E}_{x\sim p_{data}(x)}\left[\log\left(D(x|c)\right)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G(z|c)\right)\right)\right] \quad (1)$$

where the input noise variables ($p_z(z)$) and conditional variable (c) are inputs in the generator network. The real data (x) and conditional variable (c) are inputs in the discriminator network.

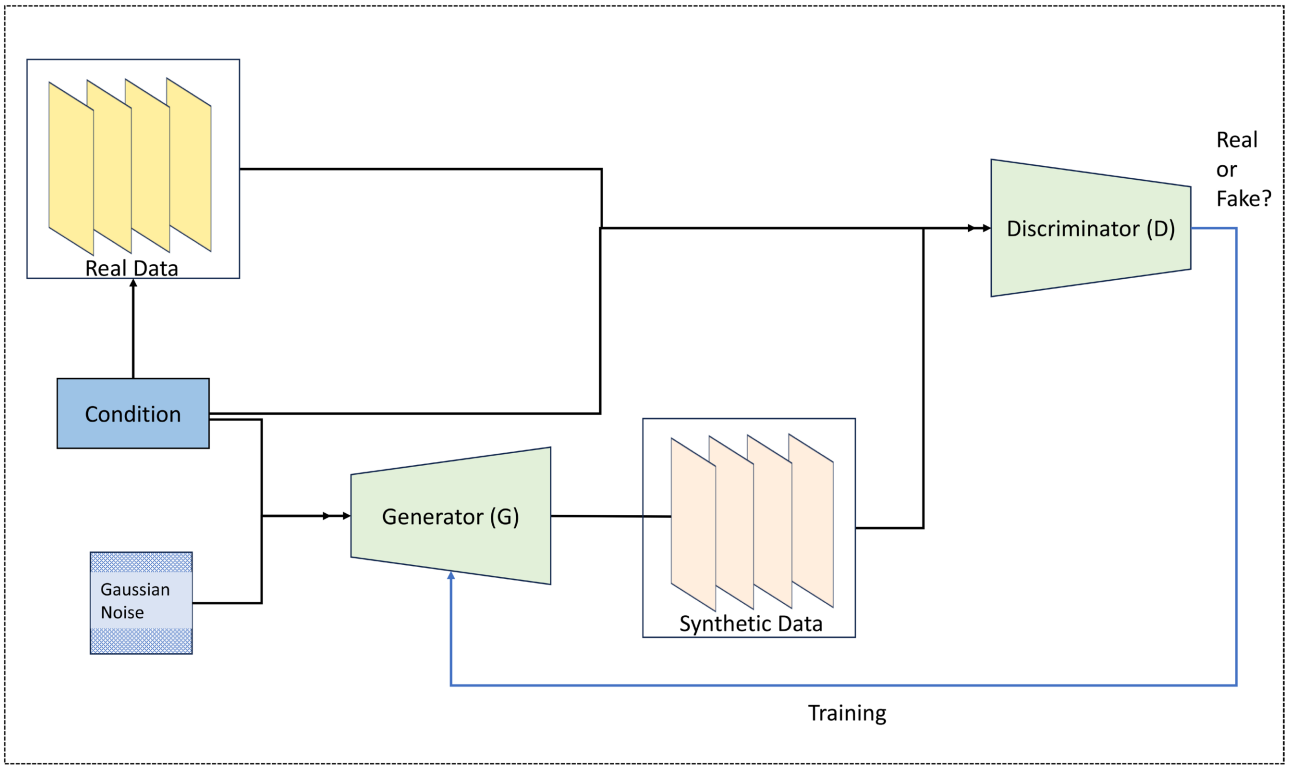

Figure 3. Architecture of a conditional GAN for synthetic data generation: real data inputs are transformed into synthetic data outputs through the generator network, conditioned on input features

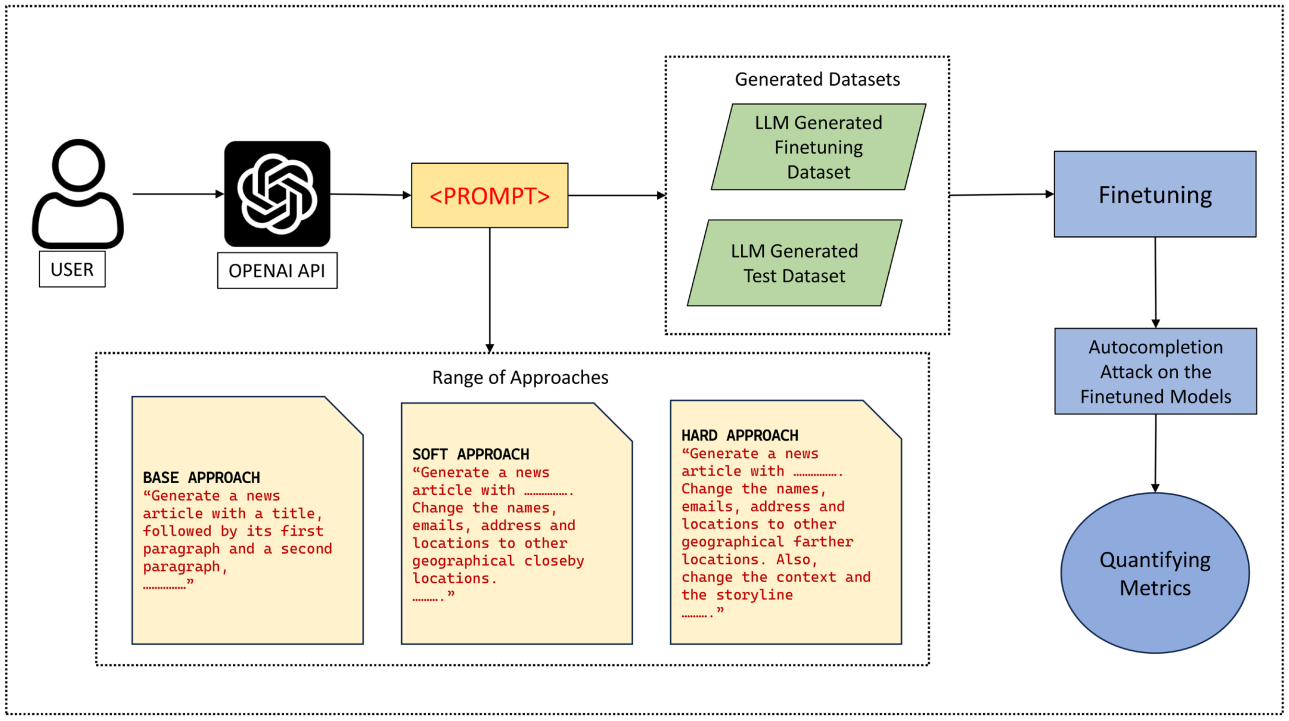

Figure 4. Overview of the workflow for LLM-generated datasets. For each specific approach and successive refinement round, a dataset is synthetically generated by querying the GPT-4 model API to produce the data points. Each particular set of training and testing data is then subjected to fine-tuning and subsequently an autocompletion attack to assess the capacity of the fine-tuned model to leak specific PII. This performance is quantified using an array of chosen evaluation metrics.

3.3.2. Dataset Generation Using LLM

The data for this subset of the study was curated using the GPT-4 model, accessed through the OpenAI API key. Each entry in the dataset adheres to a structured format, comprising a title, followed by the first and second paragraphs of an article. To guide the model's output, it was specifically instructed to produce entries in this designated format, leveraging a few-shot learning approach with a set of example prompts. At all the instances a total of 500 entries were produced for the training set and 100 for the testing set.

These datasets that are generated across multiple sets for the pair of testing and training data both by the GAN and the LLM, are subjected to finetuning in each round, referred to as successive refinement. Each round of successive refinement produces a model finetuned on its respective dataset. This process for the GAN generated can be referenced in **Figure 3** and for the LLM generated data is illustrated in **Figure 4**.

3.4. Model Utilized

The model utilized for the experiments in our study is Mistral 7b Instruct v0.2, developed by the Mistral AI team. This model incorporates advanced architectural innovations that enhance its ability to understand and generate human-like text with remarkable accuracy and coherence hence despite its size, this language model performs exceptionally well. Additionally, the model has been designed with robust security and privacy safeguards, making it a secure and reliable choice for applications that require stringent data protection measures. This model surpasses Llama 2 13B on all benchmarks and outperforms Llama 1 34B on many benchmarks. Furthermore, the Mistral 7B Instruct v0.2 model is indicative of a new era in AI development, where smaller models are increasingly capable of delivering performance that rivals or even exceeds that of much larger counterparts. This makes it an ideal candidate for testing the potential of our novel approach, as it combines state-of-the-art capabilities with enhanced efficiency.

3.5. Attacks Induced

In our study we deploy the autocomplete attack, a prevalent method among black box attacks, to pry information from the powerful large language models. This technique involves coaxing a finely tuned language model LLM to generate text or complete a given prompt. The responses are then meticulously analyzed using a suite of metrics. In one of the previous studies [15], researchers prompted a fine-tuned LLM to craft the entirety of an email based only on its subject line. Inspired by this, we challenge our LLM to generate full news articles from just their headlines. During this process, the model may inadvertently disclose Personally Identifiable Information PII embedded in its training data, and the generated content may reflect specific patterns from these datasets. Our aim is to achieve targeted forgetting of previous data and we accomplish this by repeatedly finetuning the model with different datasets, each bringing its own unique variations. Our goal

is to explore and refine methods that enable models to forget specific information while retaining their overall capabilities and utility.

4. Approaches for Dataset Generation

In this study, we employ three distinct approaches for dataset generation, each carefully designed to prioritize utility while simultaneously enhancing privacy. The data generated across these approaches adheres to a consistent structure and maintains the same division between training and testing sets. It is important to note that from the first to the last approach, the GAN or the LLM employed becomes progressively stringent in safeguarding critical data points, PII or any other sensitive information.

Each approach undergoes multiple rounds of successive refinement. During these rounds, parameters are adjusted to ensure the generation of datasets that are consistently yet diversely structured across various dimensions.

In our approach to generating data using CGANs, we have meticulously defined and adjusted several crucial parameters. The latent dimension plays a pivotal role by determining the size of the input noise vector fed into the generator, thereby shaping the diversity and quality of the generated text. Concurrently, the condition dimension governs the scope of input conditions such as class labels or textual prompts, significantly enriching the model's capability to grasp contextual nuances. Increasing the number of training epochs further hones the CGAN model's proficiency in text generation, whereas optimizing the batch size expedites data processing and facilitates convergence during training. Additionally, we have implemented regularization techniques like dropout to safeguard against overfitting, ensuring the model's ability to generalize well to unseen data. In terms of architecture, both the generator and discriminator are structured using LSTM networks, chosen for their efficacy in capturing and processing sequential data patterns.

For datasets generated using an LLM, this refinement process is controlled by manipulating a specific parameter known as temperature. The temperature parameter regulates the randomness of the LLM's output. In successive rounds of dataset generation, the temperature is incrementally increased, beginning at a value of 0.7, to strike an optimal balance between creativity and coherence. This value is subsequently raised by 0.05 in each successive round to maintain output consistency along similar lines.

4.1. Base Approach

The base approach serves as an ideal method for instructing the GAN or LLM to generate data in the format of news articles, consisting of a title, an introductory paragraph, and a subsequent paragraph.

For the base approach of generating text using CGANs, the title and other non-sensitive information serve as conditions to guide the text generation process. The generator receives these conditions along with Gaussian noise to produce new text

where only the specified sensitive details, like locations, are altered. This technique ensures that the core content and context of the original text are preserved, maintaining the integrity and coherence of the information. By conditioning the GAN on the unchanged parts of the input, we ensure that only the designated elements are varied, resulting in consistent and contextually relevant text that respects privacy concerns. In this context, the LLM was simply prompted to generate news articles in the specified format, incorporating crucial data points such as names, locations, and emails to enhance realism. It is noteworthy that, across multiple successive rounds of data generation beyond the initial dataset, the LLM was prompted to use the same headlines as in the first dataset. This practice ensures consistency in the generated data, particularly for news articles. The temperature parameter of the LLM, which controls the randomness of the generated output, was iteratively increased in each subsequent round, as previously specified.

A total of four successive rounds of datasets were generated for the base approach to assess its performance within the pipeline.

4.2. Soft Approach

This approach is primarily focused on enhancing privacy while maintaining data utility. The core idea is to preserve essential information by replacing names, email addresses, and locations present in the base dataset with dissimilar, yet contextually similar, alternatives. Specifically, for locations, this approach involves substituting actual places with nearby locations or cities. This tactic provides the model with an illusion that specific addresses or the locations of buildings and headquarters are not exact, thereby promoting privacy without losing contextual relevance. In this soft approach, all articles adhere to the same structural format as in the base dataset, consisting of a title, an introductory paragraph, and a subsequent paragraph. However, they appear dissimilar in terms of sentence formation and the handling of Personally Identifiable Information (PII) such as names, addresses, and emails.

For generating texts with CGANs, we had to understand which part would play greater role and influence the overall structure of the generated data. In the base approach, the title remains the same, but slight changes are made to paragraph 1 and paragraph 2. This can be achieved by modifying the generator model to take the title as an additional input and generating the paragraphs based on the title, while also incorporating some randomness to introduce slight change. The number of epochs and layers remain almost the same for this approach as base.

Whereas, in the context of data generated by the large language model, specifically the GPT-4 model, a specifically crafted prompt is used for this approach. The prompt explicitly instructs the LLM to read and understand the content of the base articles in the initial dataset and subsequently apply the modifications outlined in the soft approach. This ensures that the model accurately interprets the base content before making necessary alterations. In this approach, we also ensure consistency in the headlines of the generated articles to maintain contextual

integrity and prevent the model from generating random news articles. Similar to the base approach, the temperature parameter of the LLM is incrementally increased with each successive round of data generation. This gradual adjustment helps strike a balance between creativity and coherence while preserving privacy and utility.

A total of 5 sets of data across both the training and the testing sets were generated for this approach via both the GAN and the LLM.

4.3. Hard Approach

The hard approach as well is designed to safeguard private and sensitive information while preserving consistency to yield desirable results. Similar to the soft approach, new datasets are generated for each round, but with a more stringent methodology. Both the GAN and the LLM are instructed to maintain the context of the news articles by using the base headline, first paragraph, and second paragraph. However, they are required to produce a different storyline. To further protect important data points, particularly Personally Identifiable Information (PII), the data-generating systems are instructed to replace names, addresses, emails, and locations with distinctly different values. This approach ensures that while the model understands the content, it does not access exact sensitive information, thereby safeguarding it against potential security breaches that could be exploited through various attacks on any LLM. A noteworthy aspect of the hard approach is the specific handling of locations or cities. The GAN and the LLM are directed to replace these data points with distant places, thereby significantly shifting the basis of information to enhance privacy. This rigorous method ensures a higher level of protection for sensitive data while maintaining the overall utility and context of the generated datasets.

In order to generate data using CGANs, we have increased the number of training epochs batch size and adjusted the learning rate. We have also implemented curriculum training by starting with a fully guided scheme using the true previous token and gradually transitioning to a less guided scheme. Begin by only allowing the model to generate text conditioned on the true previous token, then slowly increase the probability of using the model's own generated tokens as input during training. This forces the model to learn to generate coherent text that matches the desired conditional information (e.g. title, paragraph 1, paragraph 2) from scratch, without relying on the true text. Gradually reducing the guidance from the true text over the course of training encourages the model to learn robust representations that can generate realistic text conditioned on the provided information. This hard curriculum approach can lead to more stable training, better generalization, and higher quality generated text compared to training without curriculum learning. However, it requires carefully scheduling the probability of using generated vs true tokens to ensure the model learns effectively at each stage of training. **Figure 5** shows an example of a synthetically generated data point utilizing Conditional GANs.

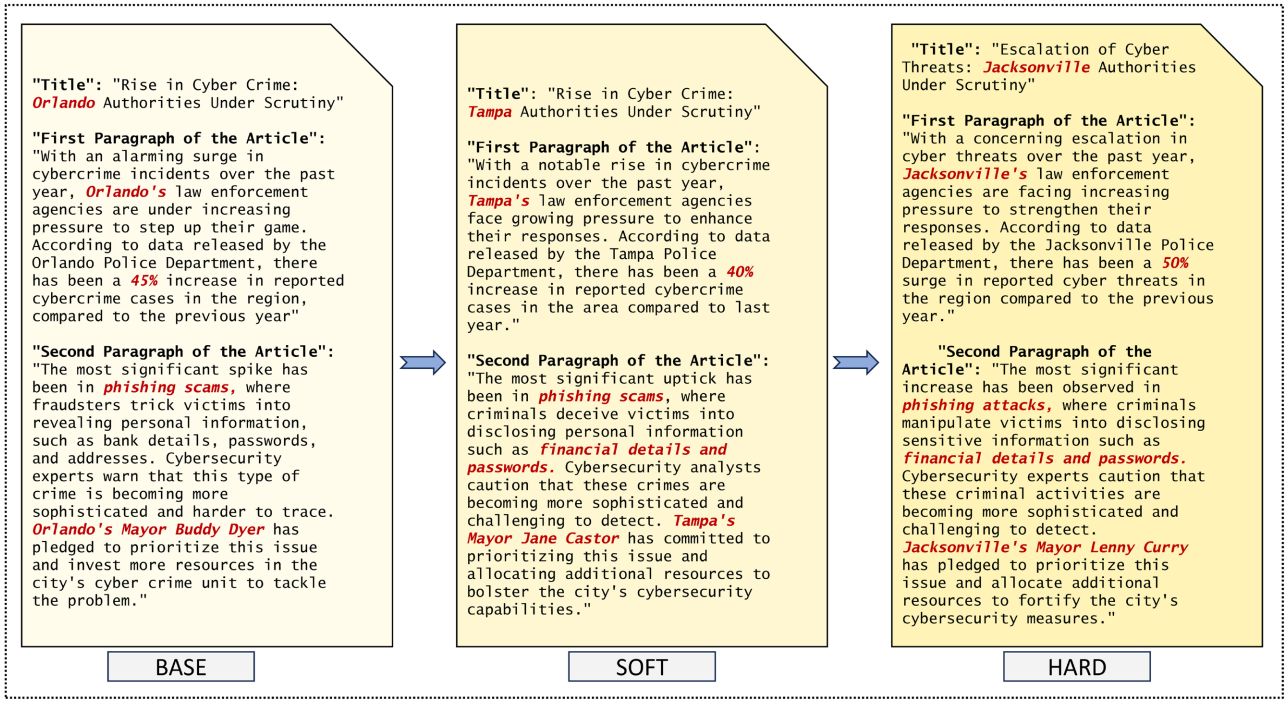

Figure 5. Example of a synthetically generated data point utilizing conditional GANs.

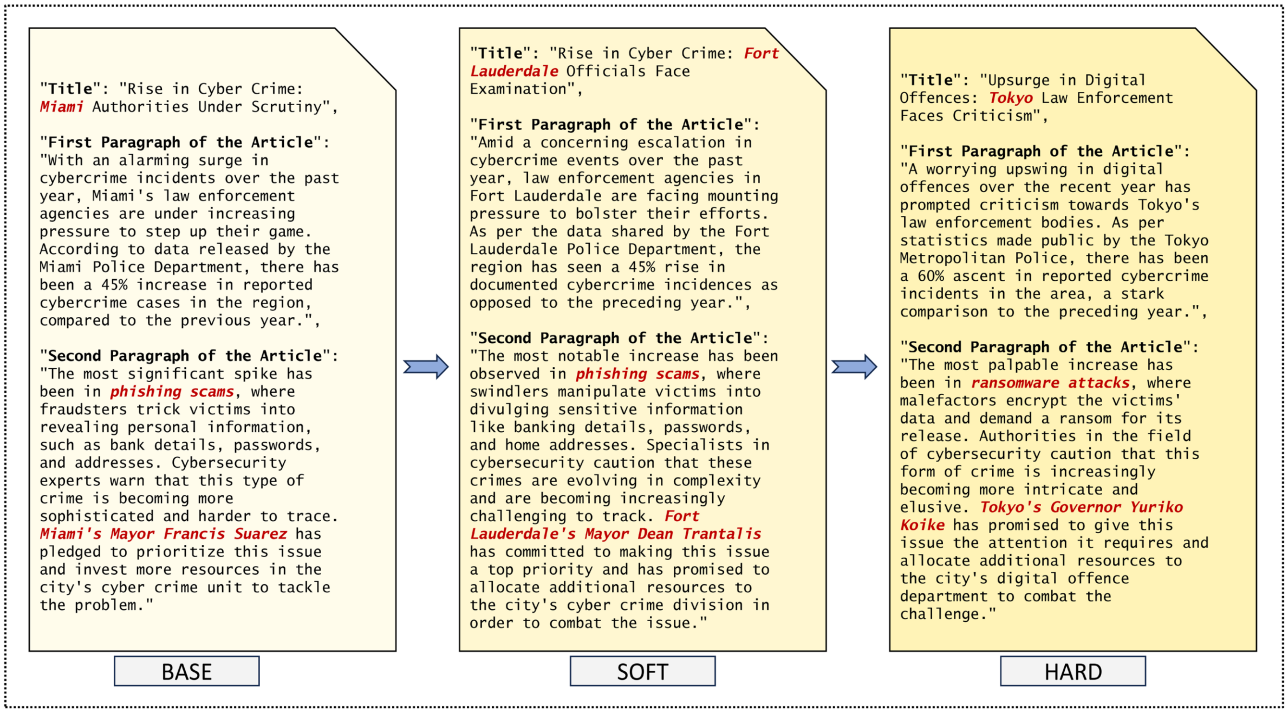

Figure 6. Example of a synthetically generated data point utilizing GPT-4 LLM.

Similar to the base and soft approaches, in the hard approach via the LLM, we prompt the model to generate the dataset. The prompt instructs the model to leverage the base data points in the form of news articles [headline, first paragraph, second paragraph] while strictly generating content that maintains the same

context but presents a different storyline. Additionally, the LLM is directed to generate dissimilar PII compared to what is present in the base dataset, with a special emphasis on changing locations to distant geographical places. As with the other two approaches, the model's temperature parameter was consistently adjusted with each successive round of data generation. This adjustment promoted coherence in the output while also ensuring variation in the generated data patterns with each new set. In this approach as well, a total of 5 sets of data across both the training and the testing sets were generated for this approach via both the GAN and the LLM. Figure 6 shows an example of a synthetically generated data point utilizing GPT-4 LLM.

5. Approaches for Finetuning

In the following section, we detail three distinct approaches—Approaches 1, 2, and 3. Each aimed at refining and enhancing the performance of the model through iterative finetuning processes. Approach 1 serves as our baseline, Approach 2 introduces a strategic shift by utilizing an already refined model from the base approach in subsequent soft and hard approaches, emphasizing enhanced contextual understanding and privacy management. Approach 3 further advances with a highly informed LLM ensuring thorough evaluation across five iterative rounds for both soft and hard methodologies.

5.1. Approach 1

The subsequent datasets (set1, set2, set3, set4) are used to fine-tune Mistral 7b, resulting in refinement models 1, 2, 3, and 4. The first approach serves as the baseline for our study. Each of the three distinct approaches undergoes four successive rounds of refinement. For both pipelines—GAN generated data and LLM generated data, the datasets used for finetuning the LLM are independently curated for each round. In the initial round of all three approaches, the pretrained Mistral 7b Instruct v0.2 model is employed for finetuning. For the subsequent rounds, the model refined in the previous round is used, creating a sequential and iterative chain-like structure. This methodology ensures a progressive enhancement in model performance and consistency in data generation and refinement. The datasets used for finetuning in both the GAN generated data pipeline and the LLM generated data pipeline are meticulously curated independently for each round. This independent curation is critical to ensure that the refinements are not influenced by the same data, allowing a clearer assessment of each approach's effectiveness. Figure 7 illustrates the working of Approach 1, which serves as the baseline experimentation for this study. In this approach, a pre-trained LLM is utilized as the foundation model for the initial refinement round. Each subsequent refinement round is conducted in a chained process, building upon the previous round. Three approaches under the TCF framework—base, soft, and hard—are employed to calculate performance and quantify the associated privacy levels. For all rounds of fine-tuning, respective datasets are generated using two distinct resources: a cGAN and an LLM.

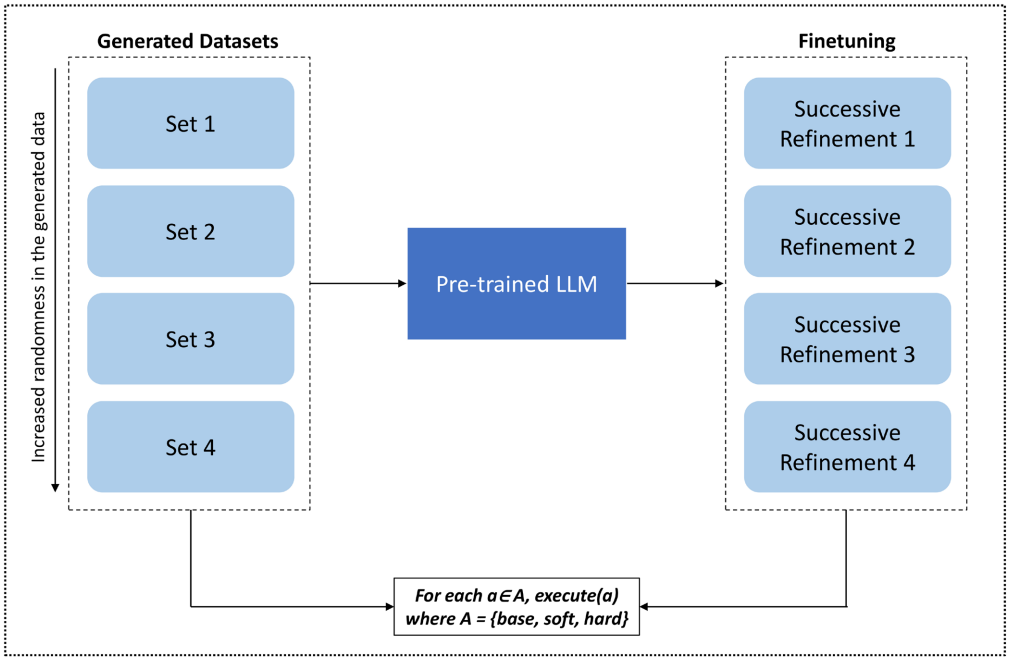

Figure 7. Working of Approach 1, which serves as the baseline experimentation for this study.

5.2. Approach 2

We utilize a more informed LLM for finetuning, aiming to enhance the model's contextual understanding and mitigate potential privacy risks. Our approach begins with the base method, which operates similarly to the baseline approach 1. But, as we transition to the soft and hard approaches, the model used for the first round of refinement is the outcome model from the first round of fine-tuning in the base approach. This strategic shift ensures that the model employed in the soft and hard approaches is already refined and contextually aware, rather than being a plain, pretrained LLM. By leveraging a model that has undergone initial refinement, we can ensure better handling of contextually relevant information. This approach allows the model to learn and integrate subtle nuances and contextual cues more effectively. In each iteration, the PII encountered by the model changes. This iterative process enables the model to overwrite multiple sets of PIIs, thereby reducing the chances of PII being leaked through backdoor or black-box attacks. We envision this process as a way to systematically tell the model contextually relevant information multiple times, ensuring that it learns to prioritize context while minimizing PII exposure. In this study, we iterate over five rounds for both the soft and hard approaches. Each round builds upon the previous one, allowing the model to integrate new information while retaining critical contextual knowledge. Figure 8 illustrates the working of Approach 2, where our approach starts with the base method, mirroring the baseline of Approach 1. As we move to the soft and hard approaches, the model for the first round of refinement is the result from the first round of fine-tuning in the base method. We conduct five rounds of iterations for both the soft and hard approaches, with each round building on the previous one.

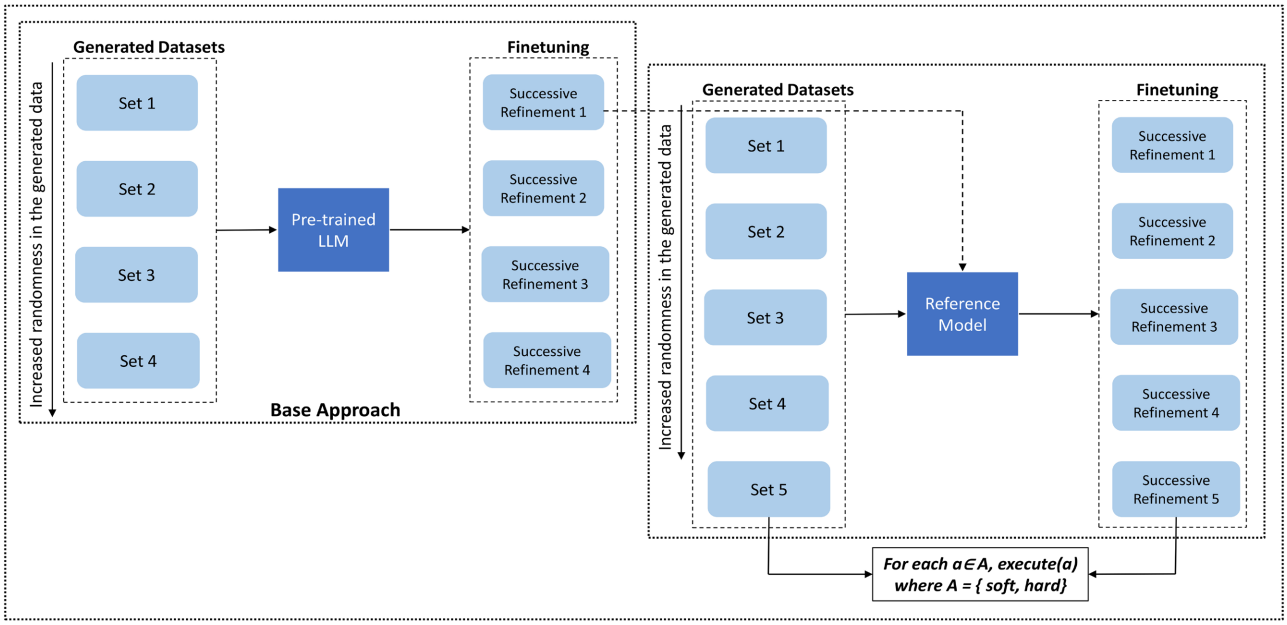

Figure 8. Working of Approach 2, where our approach starts with the base method, mirroring the baseline of Approach 1.

5.3. Approach 3

In Approach 3, we employ an even more informed LLM for the finetuning processes in both the soft and hard approaches. Similar to Approaches 1 and 2, the base approach runs concurrently; but the distinctions lie within the soft and hard approaches. Specifically, we utilize the outcome model from round 4 of the base approach, envisioning it as a model that is highly informed and contextually aware for our testing purposes.

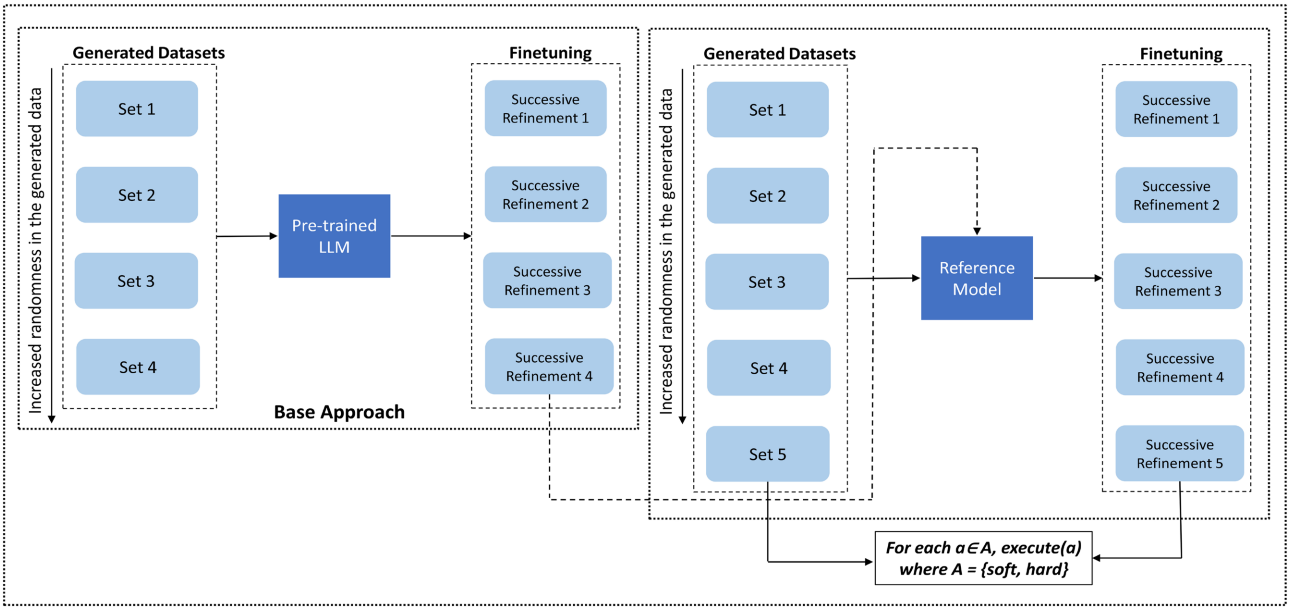

Figure 9. Working of Approach 3, we start with the base method. For the soft and hard approaches, we use the model from round 4 of the base method. We perform five rounds of iterations for both the soft and hard approaches, with each round building on the previous one.

The round 4 model is achieved by finetuning across various datasets generated by both GAN and the LLM. Initially, a pretrained LLM is used in round 1, and in subsequent rounds, the outcome model from the previous finetuning refinement is employed. This iterative process ensures that the model becomes increasingly contextually relevant and informative, surpassing the capabilities of the model used in Approach 2.

Similar to Approach 2, Approach 3 also involves five rounds of testing for the soft and hard approaches, ensuring thorough evaluation and comparison of the results. Figure 9 illustrates the working of Approach 3, we start with the base method. For the soft and hard approaches, we use the model from round 4 of the base method. We perform five rounds of iterations for both the soft and hard approaches, with each round building on the previous one.

6. Evaluation Metrics

For this study, we employ a comprehensive array of specialized metrics to quantify our results.

6.1. Extraction Success Rate

Extraction Success Rate: This metric measures the proportion of unique personally identifiable information (PII) entities extracted by the language model (LLM) from its fine-tuning dataset. It evaluates the model's susceptibility to black box attacks, specifically autocompletion attacks. Calculated using recall, it represents the percentage of all unique PII sequences present in the attacker's extracted set. Lower extraction rates indicate more effective privacy protection, highlighting the model's ability to safeguard sensitive information. This metric is essential for assessing the efficacy of privacy-preserving techniques implemented in the model, aiming to minimize privacy breaches.

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

where, "No. of unique PII sequences in attacker's extracted set common with the dataset" represents the count of unique PII sequences that an attacker has extracted and that are present in the original dataset, and "Total amount of unique PII in the dataset" denotes the total number of unique PII sequences in the entire dataset.

6.2. Cosine Similarity

We utilize cosine similarity to evaluate the similarity between two pieces of text, the base and the output generated post each successive refinement round. Mathematically, cosine similarity measures the cosine of the angle between the vector representations of each text in a multi-dimensional space. In this space, each dimension corresponds to a unique word, and the value in each dimension represents the frequency of that word in the document. Cosine similarity is particularly useful in text analysis because it provides a measure of how similar two texts are,

irrespective of their length. This is achieved by focusing on the orientation rather than the magnitude of the vectors. This score ranges from −1 to 1, where 1 indicates that the two texts are identical, 0 indicates no similarity, and −1 indicates complete dissimilarity. By applying cosine similarity, we can quantitatively compare the generated text with reference texts to evaluate how closely the model's output aligns with the expected content. This metric is integral to our analysis, as it provides a nuanced understanding of the model's ability to replicate the contextual and thematic elements of the original dataset, thereby ensuring that the generated text maintains a high degree of relevance and coherence. The formula for cosine similarity is as follows:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}, \tag{2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are vectors, $\mathbf{A} \cdot \mathbf{B}$ represents their dot product, and $\|\mathbf{A}\|$ and $\|\mathbf{B}\|$ are their magnitudes (or Euclidean norms).

6.3. Jaccard Similarity

We use Jaccard similarity to compare the similarity between two pieces of text which measures the similarity between two text documents by comparing the size of the intersection of their unique word sets to the size of their union. This metric provides a straightforward and intuitive measure of how much two sets of words overlap. The Jaccard similarity score ranges from 0 to 1, where 0 indicates no similarity (*i.e.*, no common words between the texts), and 1 indicates perfect similarity (*i.e.*, the texts have identical sets of words). By focusing on the presence or absence of words rather than their frequency, Jaccard similarity provides a clear picture of the commonality between two texts. Additionally, Jaccard similarity complements other metrics like cosine similarity, providing a more comprehensive evaluation of text similarity. While cosine similarity considers the overall thematic alignment, Jaccard similarity offers insight into the exact word overlap, giving a fuller picture of textual similarity and ensuring robust validation of our model's output. The formula for calculating Jaccard similarity is:

$$J(\text{doc}_1, \text{doc}_2) = \frac{\text{doc}_1 \cap \text{doc}_2}{\text{doc}_1 \cup \text{doc}_2} \tag{3}$$

where $\text{doc}_1 \cap \text{doc}_2$ represents the number of common elements (e.g., words, terms) between documents doc_1 and doc_2, and $\text{doc}_1 \cup \text{doc}_2$ represents the total number of unique elements in both documents combined.

7. Results

7.1. Analysis of the Results of ESR for GANs and LLMs

Based on all the obtained results as depicted in Figure 10 and Figure 11, it is evident that Approaches 1 and 3 demonstrate enhanced privacy preservation compared to Approach 2, which yields intermediary results. For GAN-generated data, both Approaches 1 and 3 provide almost consistent results. However, for LLM-

generated data, Approach 3 proves superior by offering approximately 7.8% greater privacy protection, a substantial difference. This improvement can be attributed to the multiple rounds of successive refinements in Approaches 1 and 3 compared to Approach 2. Furthermore, across all approaches, the Hard Approach generated datasets show the most superior results. For GANs, the Hard Approach enhances privacy by lowering the extraction success rate by around 6% in Approaches 1 and 2, and about 4% in Approach 3. In LLMs, these numbers are

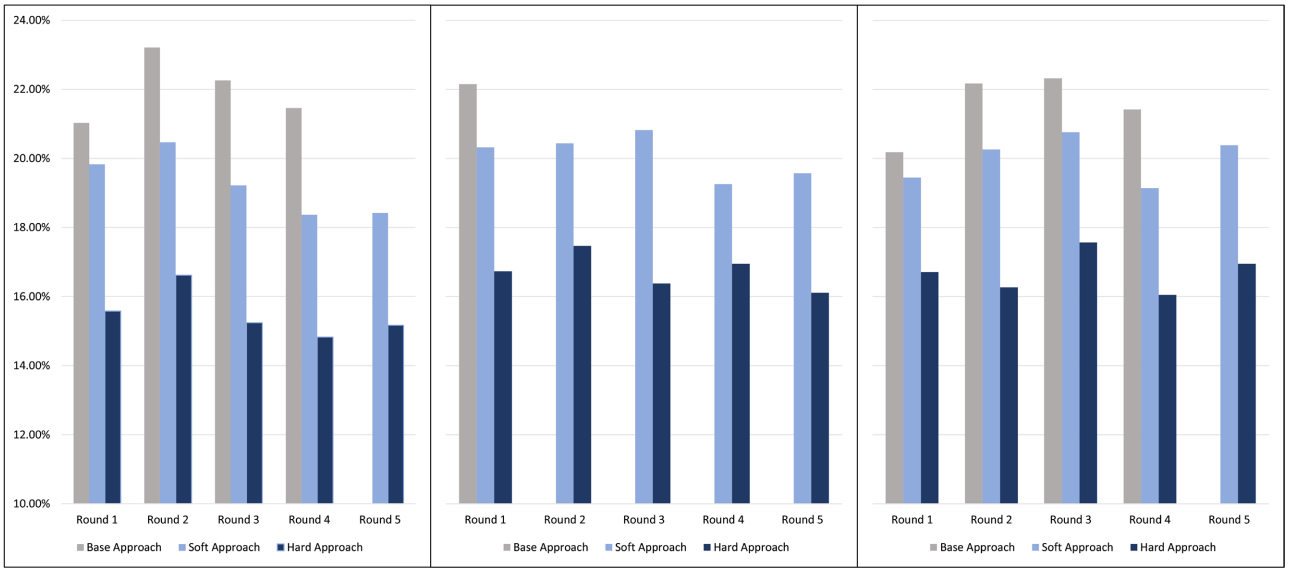

Figure 10. Graph depicting the extraction success rate for GAN-generated data across the three approaches (base, soft, hard) for each technique used to generate the data. The leftmost section represents Approach 1, the middle section represents Approach 2, and the rightmost section represents Approach 3.

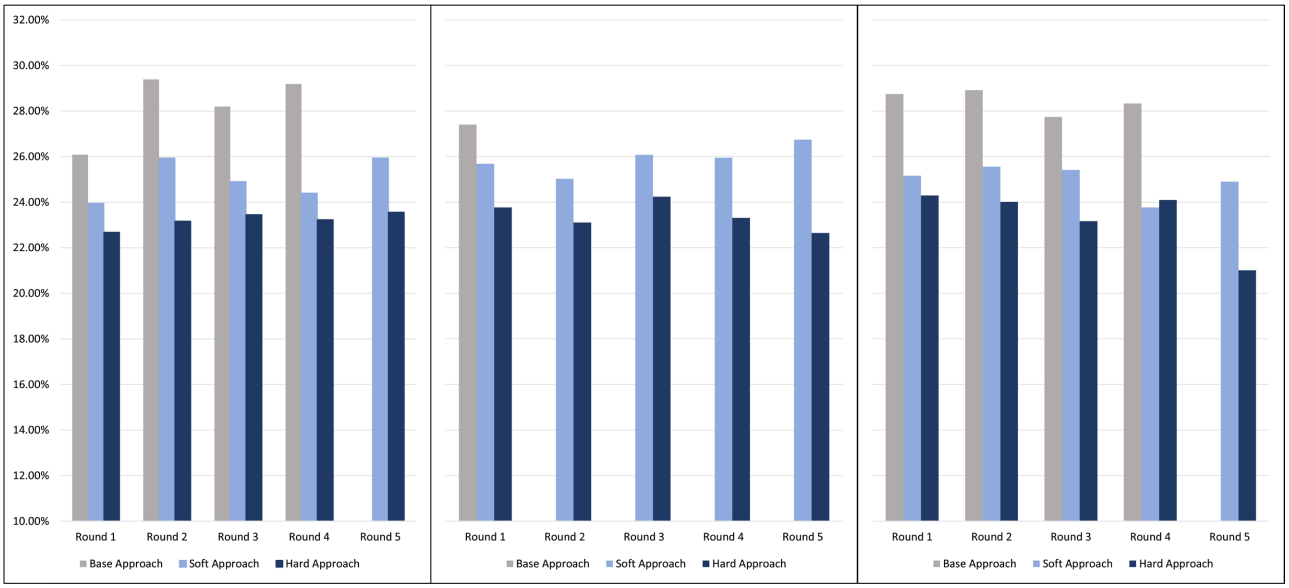

Figure 11. Graph depicting the extraction success rate for LLM-generated data across the three approaches (base, soft, hard) for each technique used to generate the data. The leftmost section represents Approach 1, the middle section represents Approach 2, and the rightmost section represents Approach 3.

even more promising from a privacy perspective, with reductions of 5.6% in Approach 1, 4.7% in Approach 2, and about 8% in Approach 3. The Soft Approach also delivers promising results, reducing the ESR value by a significant margin in GANs and peaking at a 4% reduction in Approach 3 for LLMs. This indicates that while the Hard Approach is the most effective, the Soft Approach still provides considerable privacy benefits. Lower ESR values signify improved model forgetting, which enhances privacy by reducing the risk of data extraction while preserving utility. This substantial reduction highlights the effectiveness of GANs in minimising data extraction risks. Contributing factors include the adversarial nature of GANs, which generate data that challenge the discriminator, making sensitive information harder to extract. Additionally, the use of the LLM-generated base dataset by GANs to generate the datasets likely plays a major role in enhancing privacy protection.

7.2. Analysis of the Results of Cosine Similarity and Jaccard Similarity for GANs and LLMs

The analysis of Cosine Similarity and Jaccard Similarity provides insights into the balance between privacy preservation and utility maintenance in our Targeted Catastrophic Forgetting (TCF) framework. These metrics evaluate the similarity between the generated text and the original/base text after each refinement round, across different approaches and data generation methods.

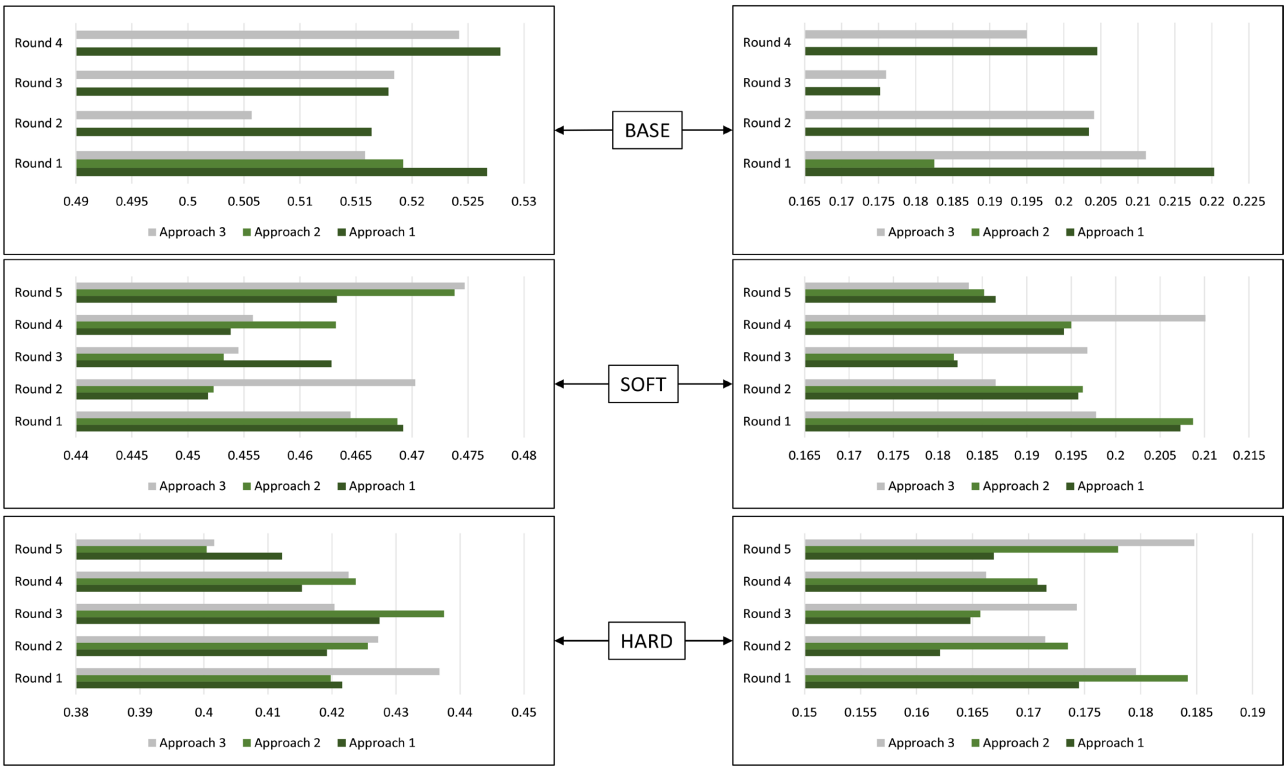

Figure 12. The above graph illustrates the similarity scores for GAN-generated data across all three fine-tuning approaches under the TCF framework. The graphs on the left display cosine similarity scores, while those on the right show Jaccard similarity scores. Both similarity measures are presented for each dataset generation approach (base, soft, hard) in the respective order.

Figure 12 illustrates the similarity scores for GAN-generated data across all three fine-tuning approaches under the TCF framework. For the base approach, the cosine similarity scores indicate a moderate level of thematic similarity, whereas the jaccard similarity indicates a lower level of exact word overlap. Both metrics show slight fluctuations across rounds, with no clear increasing or decreasing trend. For the soft approach, the cosine similarity scores indicate a lower level of similarity as compared to base approach, whereas the jaccard similarity scores are similar to the base approach. Both metrics show more stability across rounds compared to the base approach. For the hard approach, the cosine similarity and jaccard similarity scores are the lowest among all approaches. Both metrics show a slight decreasing trend across rounds, particularly in Approaches 2 and 3.

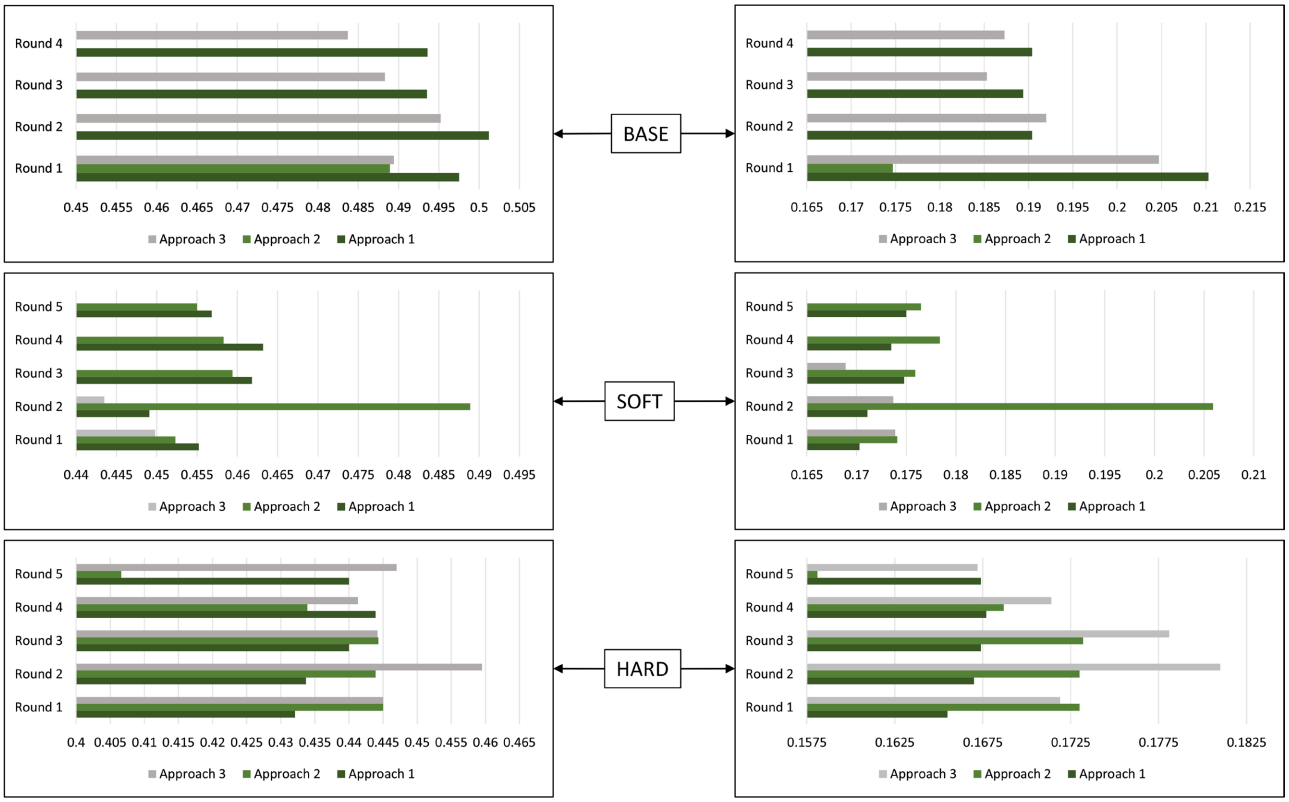

Figure 13. The above graph illustrates the similarity scores for the LLM-generated data across all three fine-tuning approaches under the TCF framework. The graphs on the left display cosine similarity scores, while those on the right show Jaccard similarity scores. Both similarity measures are presented for each dataset generation approach (base, soft, hard) in the respective order.

Figure 13 illustrates the similarity scores for LLM-generated data across all three fine-tuning approaches under the TCF framework. For the base approach, the cosine similarity scores are slightly lower than GAN-generated data, whereas the jaccard similarity scores are similar to GAN-generated data. Both metrics show more variability across rounds compared to GAN-generated data. For the soft approach, the cosine similarity and jaccard similarity scores are similar to the base approach. Both metrics show relatively stable values across rounds, with

slight fluctuations. For the hard approach, the cosine similarity scores are higher than GAN-generated data, whereas the jaccard similarity scores are similar to GAN-generated data. Both metrics show a slight decreasing trend across rounds, particularly in Approaches 1 and 3.

The results in Figure 12 and Figure 13 demonstrate the effectiveness of the TCF framework in balancing privacy and utility. Across both GAN and LLM-generated data, the Hard Approach consistently yields the lowest similarity scores, indicating the highest level of privacy preservation but potentially at the cost of utility. The Base and Soft Approaches show similar patterns, with the Soft Approach generally having slightly lower similarity scores, suggesting a balanced trade-off between privacy and utility. The choice between GAN and LLM for data generation depends on the specific requirements of thematic consistency versus word-level similarity in the target application.

8. Synthetic Dataset Generation Methods and Associated Costs

This research explores two distinct methods for generating synthetic datasets: Generative Adversarial Networks (GANs) and Large Language Models (LLMs). Both the GAN and LLM approaches are instructed to generate data within a predefined format and structure to maintain consistency across the synthetic dataset. Additionally, all generated datasets, regardless of the method used, contain the same number of data points: 500 for the fine-tuning training set and 100 for the testing set. The time taken for generating these sets is reflected in **Table 1**.

Table 1. Time comparison for synthetic data generation using CGAN and LLM approaches.

Synthetic data generation using CGAN		Synthetic data generation using LLM	
Train			
Approach	**Time**	**Approach**	**Average time taken**
Base	112 mins	Base	120 mins
Soft	82 mins	Soft	87 mins
Hard	100 mins	Hard	84 mins
Test			
Approach	**Time**	**Approach**	**Average time taken**
Base	35 mins	Base	40 mins
Soft	31 mins	Soft	17 mins
Hard	32 mins	Hard	19 mins

1) Generative Adversarial Networks (GANs):

- **Method:** We utilize a CGAN for dataset generation. A CGAN takes additional input (conditioning information) to guide the generated data towards a specific format or structure.
- **Implementation and cost:** The training process for the CGAN is conducted on Runpod with access to A40 Tensor Core GPUs. The cost of using a Cloud GPU A40 is $0.59 per hour.

2) Large Language Model (LLM):

- **Method:** We leverage the GPT-4 LLM accessed through the OpenAI API for dataset generation.
- **Implementation and cost:** The LLM generation process is conducted locally, eliminating the GPU costs associated with the GAN approach. However, utilizing the GPT-4 API incurs significant costs. We utilize the "gpt-4" model specifically, which charges $30.00 per million tokens for the input and $60.00 per million tokens for the output generated. For the scope of this study, we generated 4 base sets and 5 sets each for the soft and hard approaches. According to the pricing, it cost approximately $270 to generate these datasets.
- **Observation:** While generating the datasets using the LLM, the hard approach datasets took substantially less time to generate compared to the respective dataset generation for the base and soft approaches, with one round standing as an exception.

9. Conclusions & Future Work

In this paper, we presented a novel framework for Targeted Catastrophic Forgetting (TCF) to mitigate privacy leakage in Large Language Models (LLMs) while preserving utility. Our experiments, conducted across three distinct approaches using both GAN and LLM-generated datasets, demonstrate the effectiveness of TCF in reducing the extraction of personally identifiable information (PII). The results show that the Hard Approach within TCF consistently outperforms others in privacy preservation. The GAN-generated datasets exhibited enhanced privacy protection, likely due to their adversarial nature. The analysis of Cosine and Jaccard Similarity metrics reveals a nuanced trade-off between privacy and utility, with the Hard Approach showing the lowest similarity scores, indicating the highest privacy but potential utility loss. As evident from the results presented in this paper, TCF offers a promising technique for balancing privacy concerns with model performance in LLMs.

Currently, the TCF technique has been tested on a niche domain of news articles. This domain was selected because it offers a balance between the high prevalence of PII and coherent text, providing an adequate environment to test the TCF process. As part of future work, we will investigate the applicability of TCF in specific domains where privacy is paramount (such as medical, legal and financial), and further optimize the trade-off between privacy preservation and utility maintenance. We will explore the scalability of TCF to larger models and diverse

datasets. We will incorporate additional attacks such as Membership Inference Attack (MIA), attribute inference attacks, and more sophisticated prompt engineering techniques. Moreover, we will implement additional evaluation metrics such as Differential Privacy ϵ -value, and k-anonymity.

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., Amatriain, X. and Gao, J.F. (2024) Large Language Models: A Survey. arXiv: 2402.06196. https://arxiv.org/abs/2402.06196

[2] Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N. and Mian, A. (2024) A Comprehensive Overview of Large Language Models. arXiv: 2307.06435. https://arxiv.org/abs/2307.06435

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

[4] Pope, A. (2024) NYT v. OpenAI: The Times's About-Face. Harvard Law Review. https://harvardlawreview.org/blog/2024/04/nyt-v-openai-the-timess-about-face/

[5] Aditya, H., Chawla, S., Dhingra, G., Rai, P., Sood, S., Singh, T., *et al.* (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

[6] Chen, J. and Yang, D. (2023) Unlearn What You Want to Forget: Efficient Unlearning for LLMs. *Proceedings of the* 2023 *Conference on Empirical Methods in Natural Language Processing*, Singapore, December 203, 12041–12052. https://doi.org/10.18653/v1/2023.emnlp-main.738

[7] Pochinkov, N. and Schoots, N. (2024) Dissecting Language Models: Machine Unlearning via Selective Pruning. arXiv: 2403.01267.

[8] Bhaila, K., Van, M.H. and Wu, X.T. (2024) Soft Prompting for Unlearning in Large Language Models. arXiv: 2406.12038. https://arxiv.org/abs/2406.12038

[9] Mirza, M. and Osindero, S. (2014) Conditional Generative Adversarial Nets. arXiv: 1411.1784.

[10] Yan, B.W., Li, K., Xu, M.H., Dong, Y.Y., Zhang, Y., Ren, Z.C. and Cheng, X.Z. (2024) On Protecting the Data Privacy of Large Language Models (LLMs): A Survey. arXiv: 2403.05156. https://arxiv.org/abs/2403.05156

[11] Kumar, A., Murthy, S.V., Singh, S. and Ragupathy, S. (2024) The Ethics of Interaction: Mitigating Security Threats in LLMs. arXiv: 2401.12273. https://arxiv.org/abs/2401.12273

[12] Liu, Z.Y., Wang, J.H. and Liang, Z.W. (2019) CatGAN: Category-Aware Generative Adversarial Networks with Hierarchical Evolutionary Learning for Category Text Generation. arXiv: 1911.06641.

[13] Jabbar, A., Li, X. and Omar, B. (2020) A Survey on Generative Adversarial Networks: Variants, Applications, and Training. arXiv: 2006.05132.

[14] Luo, Y., Yang, Z., Meng, F.D., Li, Y.F., Zhou, J. and Zhang, Y. (2024) An Empirical Study of Catastrophic Forgetting in Large Language Models during Continual Fine-

tuning. arXiv: 2308.08747.

[15] 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 Personally-Identifiable Information? arXiv: 2307.16382. https://arxiv.org/abs/2307.16382